

©Copyright 2026

Kai Ji “Kevin” Feng

Designing for Interactive Systems Powered by Generally Capable AI

Kai Ji “Kevin” Feng

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:

David W. McDonald, Chair

Amy X. Zhang, Chair

Emily Tseng

Daniel S. Weld

Benjamin Mako Hill

Program Authorized to Offer Degree:
Human Centered Design & Engineering

University of Washington

Abstract

Designing for Interactive Systems Powered by
Generally Capable AI

Kai Ji “Kevin” Feng

Co-Chairs of the Supervisory Committee:
Professor David W. McDonald
Department of Human Centered Design & Engineering
Associate Professor Amy X. Zhang
Paul G. Allen School of Computer Science & Engineering

The key concerns of user experience within human-computer interaction have historically evolved alongside the technology it serves. This dissertation argues that generally capable AI—in particular, large language models and the agentic systems built on top of them—motivate the next such evolution. Specifically, I identify two new dimensions of user experience design for interactive systems powered by generally capable AI: *model behavior*, the behavioral guidelines and constraints that govern how a model responds to users; and *AI autonomy*, the extent to which an AI system operates without user involvement when tackling multi-step, long-horizon work.

I demonstrate that these dimensions can be made tractable through new design practices, tools, and frameworks. For model behavior, I first introduce POLICYPAD, a system that conceptualizes model behavior design as a prototyping practice. POLICYPAD affords iterative and collaborative authoring of model behavioral policies in conjunction with rapid testing of model behavior on sample usage scenarios. I then introduce CANVIL, a tool that extends the prototyping of model behavior to design canvases, allowing practitioners to translate their design artifacts into model behavioral specifications and vice versa. For AI autonomy, I introduce a five-level framework that frames AI agent autonomy as a deliberate design decision independent of system capability. I then introduce COCOA, an AI-powered document

editor that operationalizes agent notebooks, a new interface for human-agent collaboration. In concert, these contributions pave paths for expanding how designers, developers, and researchers craft user experiences in an era of generally capable AI.

TABLE OF CONTENTS

	Page
List of Figures	iv
Chapter 1: Introduction	1
1.1 Designing Model Behavior	4
1.2 Designing AI Autonomy	6
1.3 Thesis Contributions	8
1.4 Thesis Overview	10
Chapter 2: Background and Related Work	13
2.1 The Expanding Scope of User Experience	13
2.2 Designing Model Behavior	15
2.3 Designing AI Autonomy	21
2.4 Collaboration in CSCW and HRI	25
2.5 Summary	28
Chapter 3: PolicyPad: Collaborative Prototyping of LLM Policies	29
3.1 Introduction	29
3.2 Related Work	31
3.3 Observational Study	34
3.4 LLM Policy Prototyping	39
3.5 PolicyPad	40
3.6 Evaluation Study	49
3.7 Findings	55
3.8 Discussion	61
3.9 Limitations and Future Work	66
3.10 Conclusion	67
Chapter 4: Canvil: Designerly Adaptation for LLM-Powered User Experiences	68
4.1 Introduction	68

4.2	Related Work	71
4.3	Formative Study	72
4.4	CANVIL: a Technology Probe for Designerly Adaptation	79
4.5	Design Study	85
4.6	Results	90
4.7	Discussion	96
4.8	Limitations and Future Work	101
4.9	Conclusion	102
Chapter 5: Levels of Autonomy for AI Agents		104
5.1	Introduction	104
5.2	Terminology	106
5.3	Five Levels of Autonomy for AI Agents	108
5.4	Autonomy Certificates for AI Agents	121
5.5	Evaluating Autonomy: A User-Assisted Approach	124
5.6	Conclusion	126
Chapter 6: Cocoa: Co-Planning and Co-Execution with AI Agents		128
6.1	Introduction	128
6.2	Related Work	132
6.3	Formative Study	134
6.4	Cocoa: System Walkthrough and Implementation	139
6.5	Lab Study	148
6.6	Lab Study Results	152
6.7	Deployment Study and Results	158
6.8	Discussion	162
6.9	Limitations and Future Work	166
6.10	Conclusion	167
Chapter 7: Discussion		169
7.1	Must We Trade Off AI Autonomy and Human Agency?	169
7.2	The Evolving Relationship Between Model Behavior and AI Autonomy	170
7.3	Continual Design in an Era of Rapidly Advancing AI	172
7.4	Ethical Considerations	174

Chapter 8:	Conclusion	177
8.1	Summary of Contributions	177
8.2	Ongoing and Future Work	178
8.3	Looking Ahead	182
Appendix A:	Canvil: Supplementary Materials	183
A.1	Formative Study Participants	183
A.2	CANVIL System Prompt Template	183
A.3	Design Study Recruitment Details	184
A.4	Design Study Participants	184
A.5	Feasto User Groups	185
Appendix B:	PolicyPad: Supplementary Materials	189
B.1	Novel Policy Statements (Mental Health)	189
B.2	Novel Policy Statements (Law)	192
B.3	Themes from Formative Study Qualitative Coding	194
B.4	Findings and System Iterations from Co-design Sessions	194
B.5	Connections Between UX and LLM Policy Prototyping	197
B.6	Starter Heuristics and Policy Objectives Section	200
B.7	Post-Study Policy Rating Questions	201
B.8	Evaluation Study Participants	201
B.9	Areas of Disagreement Across Expert Groups	203
Appendix C:	Cocoa: Supplementary Materials	207
C.1	Formative Study Participants	207
C.2	COCOA Formative Study Procedure Details	207
C.3	Formative Study Data Analysis	209
C.4	Further Implementation Details	210
C.5	Lab Study Participants	211
C.6	Lab Study Recruitment Process and Pilot Studies	211
C.7	Lab Study Procedural Details	213
C.8	Lab Study Self-Evaluation Form Questions	214
C.9	Deployment Study General Guidelines	215
Bibliography		216

LIST OF FIGURES

Figure Number		Page
1.1	An overview of my contributions in this dissertation, mapped to two dimensions for user experiences powered by generally capable AI: model behavior and AI autonomy.	3
3.1	Research Process Overview. Our work proceeded in 4 phases: (1) a 15-week observational study with 9 mental health experts (19 workshops) led to (2) conceptualization of LLM policy prototyping. We then (3) designed and built POLICYPAD and (4) evaluated it through 8 policy prototyping sessions with 22 experts (10 mental health, 12 legal).	32
3.2	Timeline of activities in our 15-week observational study. During <i>taxonomy development</i> , experts organized and taxonomized a collection of diverse LLM usage scenarios. During <i>rule writing</i> , experts drafted, discussed, and refined rules to govern LLM behavior. During <i>co-design</i> , experts interacted with and gave feedback on prototypes we built of a tool for collaborative policy design. There was fluid movement between taxonomy development and rule-writing, such that some sessions included activities pertaining to both goals.	36
3.3	Illustration of our envisioned <i>LLM policy prototyping</i> process. Scenarios inform desiderata for the policy via heuristics, which in turn guide the design of the policy. The policy shapes the behavior of a policy-informed LLM, which designers can then test against the scenarios to observe changes in behavior. The process is iterative: feedback from testing may lead to the creation of new scenarios, heuristics, and policy statements.	41
3.4	Main components of the POLICYPAD system. Users can keep track of their policy version in the left sidebar (A) as they collaboratively edit the policy in the editor (B) . Users can access scenarios via the scenario gallery (C) . When they click into a scenario, they can view its full details and explore how the policy-informed model will behave on it using the scenario sidebar (D)	43
3.5	Scenarios can be brought into the editor inline with the policy as interactive widgets via referencing the scenario’s title with the ‘@’ symbol. Once in the editor, all users can click on it, view it in their scenario sidebar, and flag responses for group discussion.	44
3.6	A drafting block (directly above “Conversational tone”) can be added into the editor to draft experimental policies without affecting model behavior.	45

3.7	POLICYPAD offers two ways to test the behavior of the policy-informed model against a scenario: (1) regenerating the latest AI message, or (2) continuing the conversation.	46
3.8	Upon saving the policy via the Snapshot policy button, POLICYPAD (1) adds the policy to the version history and generates a title summarizing key changes, (2) conducts an automated heuristic evaluation of the policy, and (3) updates the latest responses to all scenarios.	47
3.9	Workflow for spotlight scenarios. (1) A user can spotlight an interactive scenario widget to expand it into a card that everyone in the editor can view. (2) The model’s response can be edited collaboratively. After saving the edited response (3) and shrinking the spotlight scenario back into an interactive widget (4) , POLICYPAD automatically analyzes the edits in the context of the existing policy and heuristics to suggest a new policy statement.	48
3.10	Comparison of Likert scale responses to policies prototyped in the system vs. baseline (left) and the novelty of policy statements prototyped in the system vs. baseline (right)	59
4.1	An overview of the CANVIL interface. A: A blank CANVIL with its property menu invoked, with some additional copies styled with pre-set color options. The property menu has options for styling (1) , model response generation (2) , and model settings configuration (3) . The <i>Main Form</i> (4) contains text input fields for structured authoring of model behavioral specifications. The <i>Playground Area</i> (5) allows users to quickly test inputs and view model outputs. B: CANVIL supports collaboration by default. Here, we see Figma users collaboratively authoring a CANVIL titled “Yoda impersonator.” Like any other native Figma object, CANVIL also supports stateful duplication.	81
4.2	The “Generate” and “Settings” options in a CANVIL’s property menu can lead to three panels that appear alongside the CANVIL itself. A: <i>Playground Mode</i> for response generation, where the input is read from and written to the CANVIL’s <i>Playground Area</i> . B: <i>Design Mode</i> for response generation, where the input is read from and written to design(s) on the Figma canvas. C: Settings for selecting and configuring the LLM.	84
4.3	The setup for an individual participant’s canvas in our study’s Figma file. A: Informational packet containing descriptions of three user groups residing in North America (primarily west coast of the U.S.), the Middle East (primarily Turkey), and Asia (primarily India), respectively. B: Starter UIs for Feasto’s 3-course meal planner with example user inputs to lower the barrier for testing model responses, along with basic UI elements such as text and buttons. C: Blank CANVILs for participants to adapt LLMs.	89
4.4	Examples of UI screens designers created for Feasto’s 3-course meal planner during the study.	103

5.1	Our 5-level framework for levels of AI agent autonomy.	109
5.2	Our proposed procedure for the issuance of autonomy certificates. The agent developer would submit their agent and an autonomy case for a particular autonomy level to a third-party governing body. The governing body would run evaluations to ensure that the agent’s behavior is consistent with the autonomy case, and issue a certificate for the level described in the case. . . .	123
6.1	An overview of the COCOA user interface. An interactive plan (A) affords human-agent co-planning and co-execution: a researcher and the AI agent can collaboratively edit the plan in the document and execute the plan steps, similar to executing code cells in a computational notebook. Steps can be assigned to the AI agent (B) or the researcher (C). The researcher can freely edit the AI agent’s outputs in an interactive sidebar (D), such as adding relevant papers that the agent did not find (E) to help steer the agent with their feedback and expertise. In this example, the first three steps of a plan to summarize methods for human feedback elicitation have already been executed, and the agent is requesting guidance from the user in the next step.	129
6.2	A user invokes the agent on a piece of text in the document by clicking on the “Invoke agent” button that appears on hover whenever text is highlighted. The agent will use the highlighted text and context from elsewhere in the document to propose a series of plans, displayed in a plan selector that appears under the highlighted text. Once the user selects a plan, it becomes fully interactive in the document.	140
6.3	If major changes are made to a step that changes the rest of the plan’s trajectory, COCOA detects this and will trigger replanning. Replanning replaces subsequent steps with ones the agent suggested for “autocompleting” the plan.	142
6.4	Users can click the output preview text beneath the plan step to access the interactive sidebar (1). By editing agent outputs in the interactive sidebar (2), users draw from their expertise to edit outputs, such as adding papers the agent might have missed (3), to guide the agent. Once edited, an indicator will appear in the output’s text preview below the step (4).	144
6.5	Once execution of the entire plan is complete, the agent inserts a plan output panel in the document with the last step’s output. This allows the user to view the plan’s results within the context of their document. The panel can be expanded or collapsed, and remains in the document even if the plan itself is collapsed for easy access.	146
6.6	Participants’ typical setup for our baseline condition. On the left, the same document editor from COCOA but without an option to invoke the agent for co-planning and co-execution. On the right, the agent is situated within a chat interface.	150

6.7	Participants' interaction activity in COCOA and the baseline within the study sessions. *** indicates a statistically significant difference ($p < 0.001$ via a paired t-test) with the closest equivalent interaction in the opposing condition. Gray areas represent spans of time where the participant was not engaging in any of our coded interactions because they were answering a question posed by the study facilitator and/or waiting on the system. Definitions for all coded interactions can be found in Section 6.5.4.	154
6.8	Participants' Likert scale ratings of steerability, ease of use, and general utility across our baseline and COCOA. ** indicates $p < 0.01$ via a Wilcoxon signed-rank test.	154
6.9	Examples from our field deployment where user and agent steps were interleaved in participants' plans. The plan steps were rewritten to redact project-specific details.	161
7.1	Three possible relationships between model behavior and AI autonomy. Left: considerations around AI autonomy remain separate from and grow faster than those around model behavior. Middle: the dimensions' relationship remains balanced. Right: model behavior consumes AI autonomy, removing autonomy considerations from the harness and folding them into the model. .	172
B.1	Version 1 of POLICYPAD: a simple collaborative policy editor with a policy-informed model in the sidebar.	196
B.2	Version 2 of POLICYPAD: a block-based editor with policy versioning and more support for structured interaction with scenarios in the sidebar. The top screen shows the sidebar in a collapsed state. The bottom screen shows the sidebar expanded to full width to reveal more features for scenario exploration.	205
B.3	Version 3 of POLICYPAD: we used the same block-based editor as Version 2 but with scenarios more closely integrated into the collaborative policy editor via interactive pill-shaped widgets.	206

ACKNOWLEDGMENTS

It truly takes a village to raise a PhD student. I am deeply grateful for the many amazing people who have supported me in my journey. It is to them that I dedicate this dissertation:

- My advisors, David McDonald and Amy Zhang, for showing me the ropes of research and HCI. You opened my mind to new angles of inquiry, sharpened my scholarly thinking, and encouraged me to pursue ambitious ideas. Thank you for your guidance and support at every step of the PhD.
- My committee members: Emily Tseng, Dan Weld, and Mako Hill, whose thoughtful feedback and discussions helped this dissertation come together.
- My mentors across many internships and institutions, especially Vera Liao, Jenn Wortman Vaughan, Ziang Xiao, Mehrnoosh Sameki, Joseph Chang, Seth Lazar, Sophie Williams, Jonas Freund, Catherine Fist, Kornraphop (Ken) Kawintiranon, and Curtis Northcutt. Your mentorship has enriched me as a researcher and a person.
- My advisors and mentors from undergrad: Arvind Narayanan, Marshini Chetty, David Reinfurt, and Kevin Lee, for believing in me early on and whose guidance and wisdom I still carry with me today.
- My mentees across research projects and DRGs, especially Max Coppock, Amodini Khade, Tony Li, Lawrence Tan, Tanjal Shukla, Amber Liu, Gloria Guo, Jane Guo, and Samuel Lim. You all make research mentorship so rewarding.
- My collaborators on papers in this dissertation, who helped shape our work with their insights, care, and craft: Tzu-Sheng Kuo, Quan Ze (Jim) Chen, Inyoung Cheong,

Ken Holstein, Teddy Lee, Tyna Eloundou, Kevin Pu, Matt Latzke, Tal August, Pao Siangliulue, Jonathan Bragg, Dan Weld, Joseph Chang, Vera Liao, Ziang Xiao, and Jenn Wortman Vaughan.

- The countless participants in my studies. You make my work possible and remind me of why I chose to pursue HCI in the first place.
- My labmates in the Social Futures Lab: Quan Ze (Jim) Chen, Ruotong Wang, Galen Weld, Leijie Wang, Jina Yoon, Alicia Guo, Teanna Barrett, Matthew Bilik, Vinay Koshy, and Peipei Nie. I can't think of a better group of people to go through grad school with. I'll treasure memories of our adventures in and outside of the lab.
- My HCDE PhD cohort, who have been there with me since day 1: Adam Hyland, Brett Halperin, Danli Luo, Julie Vera, Kimberly Perkins, Laura Jo Swartley, Leslie Coney, Murtaza Ali, Nathaniel Mengist, Ruican (Rachel) Zhong, Tricia Aung, Georgia Kenderova, Samuel So, Sayan Bhattacharjee, and Zarine Kharazian.
- Staff of the HCDE and CSE departments, who make everything run smoothly so students can do their best work, especially Kathleen Rascon, Leah Pistorious, Jane Skau, Stacia Green, Allen Lee, Chris Nagle, Joe Eckert, and Elise Dorough.
- Friends in the academic community, especially Rock Pang, Kevin Pu, Raymond Fok, Alice Gao, Jacqueline He, Michelle Lam, Tony Li, Tzu-Sheng Kuo, Wesley Deng, Anna Kawakami, Sebastian Santy, Luke Thorburn, Stephen Casper, Yoonjoo Lee, and Tae Soo Kim. You all make grad school fun and memorable. Special shoutout to Rock for being my roommate for 4 consecutive CHIs.
- Friends outside academia, across grad school, college, and high school, who I know I can always count on, especially Ben, Roger, Doug, Rachel, Zach, Victor, Max, Annie, Sung, Homer, and Yiwei.

- My roommates at Blue House: Scott Martin, Tommy Nguyen, and David Alexander, for the countless meals, lively dinner conversations, and adventures around the PNW. Our cooking can be just as innovative as our research.
- My extended family, including my cousin Wendy and aunts, uncles, and grandparents in China, for your curiosity in my work and patience as I navigate what seems like an unfathomable number of years in school.
- My partner, Kate, for her love and support. I'm so lucky to have you in my life.
- My parents, Hong Fang and Wei, for your love, care, and all that you've done to provide a nurturing environment in which I feel supported in whatever I do. You've built the foundation that allows me to pursue my dreams.

The work conducted in this dissertation was supported by a UW Herbold Fellowship and an OpenAI Democratic Inputs to AI grant. Thank you for making this research possible.

Chapter 1

INTRODUCTION

Over the past half-century, the notion of *user experience* in the context of interactive software was defined through multiple cycles of evolution spurred by technological advances. In early industrial and military settings, human factors engineering focused on optimizing efficiency and minimizing operator errors [569]. The rise of personal computing in the 1980s brought about a cognitive turn: researchers shifted attention to users’ cognitive models and how they impact system *usability* [278, 468], which laid the intellectual foundations for user-centered design [469]. As the internet connected millions of new users in the 1990s, new theories and applications for computer-mediated interaction between networks of people gave rise to social computing systems, where user experience was dictated not only by the intrinsic properties of the software itself, but also other people using that software [3, 229, 258]. By the 2000s, as mobile and web applications became ubiquitous, the notion of “user experience” expanded holistically to include factors that strengthen users’ emotional connections to interactive systems, including aesthetics and values, which some argue have a greater influence on user satisfaction than practical utility alone [467].

Since the introduction of generative artificial intelligence (AI) to the public in late 2022, user experience faces yet another inflection point. AI has become *generally capable*: exhibiting capable performance across a wide range of challenging benchmarks [34, 78]. This was possible thanks to architectural innovations, such as the transformer [655], combined with training at unprecedented scale [476]. Together, they unlock fundamentally new behaviors in AI, including conversing in fluent natural language, performing multi-step reasoning on open-ended problems, and even taking actions autonomously with limited human supervision [280, 312, 707]. The step-changes in capabilities dramatically expand what interactive systems are capable of, and consequently, open up new possibilities for the user experience of AI-powered systems.

How can designers and developers of human-centered technology realize these possibilities? Traditionally, when HCI talks about crafting a user experience, the main lever we reach for is *user interfaces*: screens, interaction flows, and information architectures. Yet, this was never the whole picture. For example, in social computing systems (e.g., collaborative document editors, social media platforms), the user experience of one individual is dictated by the behavior of other users in the same system. Designing the user experience therefore involves carefully reasoning about interventions to nudge human behavior in particular directions. For example, introducing upvoting and downvoting mechanisms on platforms like Reddit or Stack Overflow incentivizes concise, informative responses while penalizing spam [449]. Generally, new technological waves prompt us to consider how we might repurpose our existing levers for user experience, and which entirely new ones will emerge.

I discuss two new design dimensions for user experience exposed by generally capable AI. The first is **model behavior**: what behavioral guidelines should the model follow? What safety constraints does it respect? How does it handle high-stakes situations with real consequences for users? The second is **AI autonomy**: what should the AI system take on, and what should it leave to its human operators? How is this distribution of labor configured and negotiated over time? These dimensions, just like UI affordances and visual aesthetics, are levers for designers to steer the development of interactive systems.

Concretely, I define model behavior and AI autonomy as follows:

- **Model behavior** describes how a model responds to user inputs. It is commonly defined via human-readable specifications, which include behavioral guidelines, constraints, and desiderata. It is often realized via post-training pipelines such as reinforcement learning from human feedback (RLHF).
- **AI autonomy** is the extent to which an AI system is designed to operate without the user’s involvement. This is characterized by the distribution of initiative, responsibility, and control between the user and AI. It can be implemented by mechanisms that decide task allocation between the human and AI, and how flexibly that allocation can shift based on context and time.

These two dimensions are not strictly orthogonal, as model behavior can certainly influence and thus be considered a component of AI autonomy.¹ Technically speaking, however, there is a clear separation of concerns: model behavior is a property of the model itself, while AI autonomy is implemented by scaffolding around the model (i.e. the harness) [137]. This relationship is visualized in Fig. 1.1. In this dissertation, I treat these dimensions as distinct by focusing on their unique aspects throughout my work. I later unpack the relationship between these dimensions in Chapter 7.2.

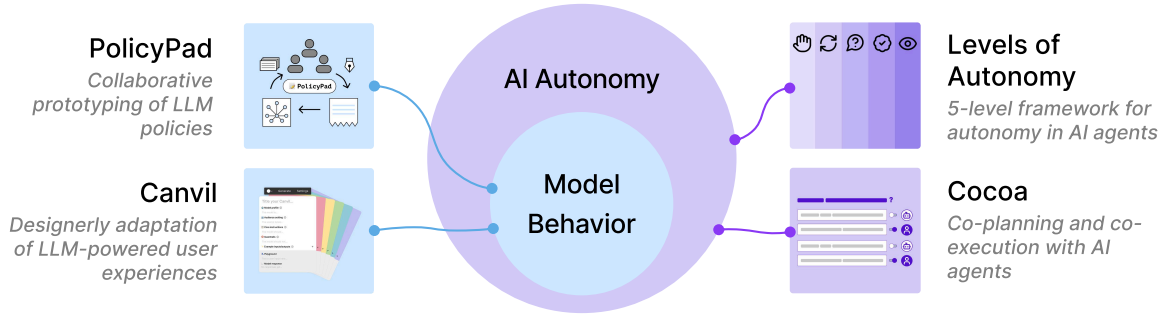


Figure 1.1: An overview of my contributions in this dissertation, mapped to two dimensions for user experiences powered by generally capable AI: model behavior and AI autonomy.

This dissertation contributes computational workflows and tools that allow designers² to exercise these two dimensions. Through my contributions, I show that these dimensions present promising levers for researchers and practitioners when building, deploying, and studying interactive systems powered by generally capable AI.

This dissertation first explores a novel design challenge—shaping model behavior—against a familiar methodological backdrop of rapid prototyping. By observing how domain experts collaboratively critique and define behaviors for LLMs, I argue that model behavior should be *prototyped* through tight, iterative feedback loops that allow practitioners to rapidly draft, test, evaluate, and refine behavioral specifications. I then introduce two systems that

¹For example, a model trained to take on an independent and confident persona might choose to not call tools that allow it to consult the user for feedback while tackling a complex task.

²I use the term “designer” broadly, referring to anyone who has a role in designing an interactive system. This can include researchers, engineers, and non-technical practitioners who steer system behavior. Chapter 4 uses a narrower definition of designer specific to that chapter.

support different environments: POLICYPAD, a collaborative editor for groups of users to rapidly prototype LLM policies; and CANVIL, an extension of the popular collaborative design tool Figma that lets designers work with real model behavior as a “design material.”

I then transition to AI autonomy, a dimension orthogonal to model behavior in AI agents—compound AI systems that can tackle open-ended tasks over extended time horizons [312]. I first show that AI autonomy can be a deliberate design decision, independent of agent capability, and then introduce a framework for AI agent autonomy. I then explore how interactive systems ought not to be constrained to a single level of autonomy, but flexibly move across them. I propose *agent notebooks* as a new interface for human-agent collaboration, and investigate their effectiveness by integrating them into COCOA, a document-based system for agent-assisted scientific research. COCOA illustrates how intentionally designing for AI autonomy opens the door for innovation in human-AI interaction paradigms.

1.1 *Designing Model Behavior*

Given two identical interfaces, two AI-powered systems can have vastly divergent user experiences due to differences in underlying model behavior. This is dictated by factors specific to the model, such as what values a model reflects, what tone it defaults to when responding to users, and what its response strategy entails when answering sensitive or high-stakes queries. These factors are all questions that AI developers have tried to make concrete through their model behavioral policies (e.g., Anthropic’s Constitution [27] and OpenAI’s Model Spec [491]). Model behavioral policies are aspirational artifacts that detail the principles, rules, and desiderata that govern how a model should behave. Anthropic’s Constitution also includes detailed rationale to justify that behavior.

Little is publicly known about the process by which these policies are created, but it is clear that their iteration cycles are by no means short. Anthropic’s Constitution, for example, was first officially published in September 2023 [18] and was only updated again in January 2026 [27]. Moreover, while there has been significant interest in including domain experts and the general public in the development of these policies [268, 475, 493, 500], there remains sparse scientific investigation into the best practices and tooling to do so.

Informed by observing how domain experts collaborated with OpenAI to contribute to

the initial version of the Model Spec, I propose for model behavior to be *prototyped*, rather than merely *defined* through a linear, one-shot process. What can processes for this new form of prototyping look like, and how can they be supported in practice?

1.1.1 Collaborative and expert-led prototyping of model behavior

Prototyping is a foundational practice in HCI that lets designers iterate cheaply and quickly experiment with ideas before committing to them. Model behavior design can also benefit from this philosophy. POLICYPAD is a collaborative system for *LLM policy prototyping*. It integrates authoring of model behavioral policies, usage scenario exploration, and model behavior testing into a real-time collaborative document editing environment. Users can fluidly move between these activities. Through tight feedback loops characteristic of prototyping, users inspect scenarios to understand current model behavior, draft policies to improve it, and collaboratively verify policy outcomes.

As with any realtime collaborative system, one concern is ensuring that synchronous activity does not overwhelm users and hinder collaboration. The design of POLICYPAD considers this by offering both shared and private workspaces. Users can collaboratively edit the policy in a shared document editor, while independently probing policy-informed model behavior on diverse usage scenarios. In POLICYPAD, the boundary between shared and private workspaces is permeable: the system allows experts to bring scenarios from independent exploration directly into the shared editor as interactive widgets, making model behavior a visible and shared reference point during collaborative prototyping. These widgets allow users to collaboratively edit the model’s response in the editor alongside the policy itself. Additionally, POLICYPAD can automatically suggest new policy statements from edited responses by analyzing changes to steer the model towards the edited behavior.

1.1.2 Integrating model behavior prototyping into user experience practice

POLICYPAD demonstrates that collaborative LLM policy prototyping can be an effective practice for non-AI experts to surface novel and relevant desiderata for model behavior. The resulting behavior from policies created in POLICYPAD can be tailored for a particular

domain if users all share a domain of expertise, or for general use. However, models are often integrated into specific applications with an established set of users and usage contexts.

CANVIL extends model behavior prototyping into professional user experience practice so designers can tailor LLM-powered user experiences based on their existing design constraints. Specifically, it supports *designerly adaptation*—a process for translating design requirements into model behavioral specifications, and vice versa. Through structured authoring of behavioral specifications, testing on diverse user inputs, and integration of model outputs into interface designs, CANVIL enables designers to prototype model behavior within their existing design workflows. In practice, designers used CANVIL to iteratively co-evolve their designs with model behavior, and to recognize the collaborative potential of designerly adaptation within their teams.

1.2 Designing AI Autonomy

As AI systems become more *agentic*—operating over long durations and taking actions via tools under limited human supervision—user experience concerns expand beyond turn-by-turn model behavior. AI autonomy thus becomes a key design consideration. Autonomy is often used as a proxy measure for AI capability advancement [426]. I argue that in interactive systems, autonomy is a design choice that can be actively calibrated independent of capability. A growing body of work recognizes that appropriately calibrating AI autonomy is a first-class design concern—not just for usability, but for safety, trust, and human agency [275, 450, 515, 588, 595]. Indeed, risks can arise across the autonomy spectrum. On one end, fully autonomous agents sacrifice human control, making it difficult for users to actively anticipate or correct the agent’s actions when needed. On the other, frequent and rigid checkpoints preserve oversight but can result in frustrating or even risky user experiences where users are primed into rubber stamping agent actions with limited thought.

Exposing AI autonomy as a new design dimension and enabling designers to work with it tractably requires both conceptual frameworks for thinking about AI autonomy, and practical interaction paradigms that bolster human-agent collaboration. I now describe two contributions that this dissertation provides along these lines.

1.2.1 *Levels of autonomy for AI agents*

Broadly, *agency* refers to the capacity to formulate an intention and act according to it [55, 616]. An AI agent’s agency is thus shaped by the environment it operates in and the tools it has access to. Autonomy³ is one central dimension of agency, referring to the extent to which an agent is designed to operate without user involvement [52, 323].

Current agent development tends to treat autonomy as a natural consequence of improving capability [312]. I show that an agent’s level of autonomy can be treated as a deliberate design decision, separate from its capability and operational environment. I define five levels of escalating agent autonomy, characterized by the roles a user can take when interacting with an agent: operator, collaborator, consultant, approver, and observer. Within each level, I outline the ways by which a user can exert control over the agent, along with open questions for how to design the nature of user-agent interaction. I also highlight a potential application of the framework towards AI autonomy certificates to govern agent behavior in single- and multi-agent systems, as well as paths forward to evaluating AI autonomy. This framework is intended to help designers and developers better reason about the appropriate levels of autonomy at which their agents should operate.

1.2.2 *Agent notebooks for human-agent collaboration*

One goal of the Levels of Autonomy framework is to map deployed agentic AI systems to a particular level. Another is to invite designers to consider how future systems can *flexibly and appropriately move across different levels*, rather than being confined to one level.

To explore this, I introduce *agent notebooks* as a new interface for human-agent collaboration. Taking design cues from computational notebooks, agent notebooks allow a user and an agent to collaboratively edit a plan of action, assign steps to either party, execute steps in sequence or selectively, and fluidly transition between planning and execution as the task evolves. This paradigm supports flexible delegation of labor between users and agents. Users may assign a higher portion of the work to the agent and have the agent operate more

³While *shared autonomy* has been a long-standing topic of interest in human-robot interaction [198, 465], a similar level of attention to autonomy has yet to be given to software-only AI agents at the time of writing, perhaps because reliable autonomous action-taking was not possible until recently.

autonomously, or take on a more active role themselves and constrain the agent’s autonomy. I implement agent notebooks into COCOA, an agent-powered document editing environment for scientific researchers. I show that COCOA improves agent steerability without sacrificing ease of use when compared to a traditional chat-based interface, and can support a range of collaboration patterns between humans and AI agents over extended use.

1.3 Thesis Contributions

This dissertation’s core contribution lies in making two design dimensions tractable: model behavior and AI autonomy. It does so through workflows, frameworks, and tools for designers to exercise these dimensions when crafting the user experience of systems powered by generally capable AI. This section summarizes the contributions of this dissertation across three themes.

1.3.1 Formalizing the practice of model behavior design

Over the past couple decades, user interface design has been formalized as a practice with distinct methods (e.g., prototyping), processes (e.g., rapid iteration, designer-developer hand-off), artifacts (e.g., wireframes), and tools (e.g., Figma). In this dissertation, I make initial steps towards formalizing a practice of model behavior design. Doing so requires methods, processes, artifacts, and tools through which practitioners can work with model behavior.

At the method level, I draw from long-established practices of prototyping in HCI to demonstrate how model behavior can be *prototyped* through iterative feedback loops of behavioral policy specification and model experimentation. Rather than treating a behavioral policy as the final word on how a model should behave, practitioners can treat it as a malleable artifact through which they experiment, challenge, and deliberate with peers about (un)desirable model behavior against representative usage scenarios.

At the process level, *LLM policy prototyping* and *designerly adaptation* both embody values of rapid, interactive experimentation and evaluation central to many design communities. They encourage practitioners to situate model behavior in their domains, evaluate it on the scenarios they care about, identify where generally capable AI falls short, and iteratively envision and test behavioral changes. These processes are collaborative: domain

experts integrate expert knowledge into behavioral policies, while user experience designers co-evolve model behavior and user interface within their native design workflows.

At the artifact level, LLM policies and behavioral specifications provide easily understandable and editable representations of how a model should behave. Realistic usage scenarios and existing design specifications guide prototyping choices and provide design constraints. These representations give practitioners shared reference points when prototyping collaboratively and iterating in response to observed model outputs.

Finally, at the tool level, POLICYPAD and CANVIL operationalize these methods, processes, and representations in interactive environments. POLICYPAD supports collaborative policy authoring, scenario exploration, and model behavior testing in a multiplayer document editor. CANVIL brings live model behavior into design canvases, allowing designers to work with models as a design material alongside UI affordances.

Altogether, these contributions form the foundations for a broader practice of model behavior design that practitioners can adopt and extend as generally capable AI permeates new domains and applications.

1.3.2 *Autonomy as a lever for improving human-AI collaboration*

In agentic AI systems, the model behavior is only one contributing factor to the user experience. Interactions between the model and agent harness⁴ can determine the frequency and means with which an agent collaborates with a human user or other AI agents.

I argue that autonomy, in the context of user experience, need not be coupled with capability. That is, just because an agent *has the ability* to work autonomously for an extended period of time, it does not imply that operating at that level of autonomy is in the user’s best interest. Indeed, recent studies suggest that excessive delegation to AI can result in skill atrophy in humans [366, 394, 594]. If a user’s goal is to maintain or even improve a skill when working with AI, the autonomy level of the AI should be calibrated accordingly. This naturally opens up new questions: how can we define these different levels of autonomy,

⁴A harness for an AI agent is the software wrapper around a model that enables it to act within its environment, besides simply responding to prompts. The harness connects a model to tools, prompts, memory, and other software like code executors [137].

and how can this calibration be done?

To do this, I first introduce a conceptual framework for reasoning about AI autonomy, characterized by 5 roles a user can take when interacting with an AI system. I then introduce COCOA, a system that implements agent notebooks as a new interface for interleaving human-AI collaborative planning and task execution in the context of scientific research. These contributions operate at complementary scales: the framework helps designers and developers reason about how much autonomy an agent should act with, while COCOA demonstrates an interaction pattern for setting and adjusting the autonomy as work unfolds. They both treat autonomy as a configurable relationship between human and agent, rather than a fixed downstream effect of model or agent capability.

1.3.3 An expanded conceptualization of user experience for generally capable AI

This dissertation encourages an expansion of the scope of user experience for generally capable AI across two distinct design dimensions: model behavior and AI autonomy. Drawing on observational, formative studies, and system contributions across multiple domains, I show that non-AI domain experts both desire and assume responsibility for ensuring effective and responsible deployment of AI into their domains and society more broadly. Furthermore, I demonstrate that as AI transitions from turn-based models to autonomous agents, operating at the model layer alone fails to address user experience concerns at the system level. Addressing this gap motivates treating AI autonomy as an explicit design dimension supported by new conceptual models and interaction techniques. All in all, this dissertation invites researchers and practitioners dedicated to advancing AI user experiences to explore, leverage, and expand upon these design dimensions in their work.

1.4 Thesis Overview

This dissertation will proceed as follows. In Chapter 2, I first position my work in the context of related research. I briefly trace the history of what “user experience” entailed since the dawn of computing to illustrate its expansion across different technological waves, and then review research relevant to designing model behavior and AI autonomy, the two design dimensions around which the remainder of this dissertation is organized.

In Chapter 3, I present POLICYPAD. I detail a 15-week observational study conducted in partnership with OpenAI, in which mental health experts discussed, annotated, and drafted LLM responses and policies across 19 workshops. I discuss how the study surfaced the need for tight, collaborative feedback loops in policy design, and motivated the conceptualization of LLM policy prototyping. I give an overview of the system design of POLICYPAD, followed by evaluations with 22 domain experts across mental health and law. I show that POLICYPAD’s design fostered collaborative dynamics between experts and yielded more novel policies compared to a baseline reflective of existing systems.

I then transition to applying a similar prototyping approach used in POLICYPAD to modern user experience practice in Chapter 4 to help designers contextualize and adapt model behaviors within specific applications. I present a formative interview study with 12 designers experienced in crafting LLM-powered user experiences, which identified the need for designerly adaptation, a translational process between design requirements and model behavior. I then introduce CANVIL, a Figma widget operationalizing designerly adaptation, and study its use through a task-based design study with 17 practitioners. Using CANVIL, designers iteratively translated design requirements into behavioral specifications, co-evolved their designs with model behavior, and recognized the collaborative potential of designerly adaptation within their teams.

In Chapter 5, I shift focus to address the increasing agency and autonomy of AI. I introduce a five-level framework to support designers and developers in reasoning about AI autonomy as a design decision, and discuss its implications for AI development and governance. The chapter also proposes autonomy certificates as a governance mechanism for communicating an AI’s autonomy-relevant characteristics, and discusses formative ideas for approaches to evaluate AI autonomy.

Chapter 6 examines how we can design for AI autonomy that can be interactively configured in practice. I first describe a formative study with 9 researchers that motivates the design of interfaces that support flexible delegation of work between humans and AI agents. I introduce COCOA as a system that implements agent notebooks as a new interaction paradigm for interleaving human-AI collaborative planning and task execution for scientific research tasks. I show the promise of COCOA through a lab study with 16 researchers,

where participants made frequent use of interleaving co-planning and co-execution, and a 7-day field deployment with 7 researchers, where participants used self- and AI-assigned steps to flexibly delegate agency and embed their expertise into the workflow.

Chapter 7 synthesizes findings across the four projects. It reflects on the relationship between AI autonomy and human agency, as well as the relationship between the two design dimensions around which this dissertation is centered. Additionally, I discuss ethical implications for designing and deploying advanced AI systems. I conclude the dissertation in Chapter 8 by reviewing its core contributions and charting new directions through ongoing and future work.

Chapter 2

BACKGROUND AND RELATED WORK

This chapter situates my dissertation within relevant work in HCI, CSCW, AI, and human-AI interaction. I first trace how the scope of user experience has expanded alongside successive technological waves, before reviewing relevant research on designing model behavior and AI autonomy.

2.1 The Expanding Scope of User Experience

User experience has always evolved alongside the technology it serves. Each major technological wave has shifted how researchers and practitioners set goals and priorities for their design practice [230].

- **Human factors and ergonomics (1950s–70s).** In industrial and military settings, the user was understood as a machine operator who needed to establish a physical and cognitive fit with complex control systems [195, 569]. Interfaces were mechanical (e.g., control panels) or command-line, and the design goal was productivity: minimize operator errors to maximize output through machine use.
- **The cognitive turn (1980s).** Personal computing shifted attention to users' mental models and how they shape system usability [278, 469]. Card, Moran, and Newell formalized interaction as information processing [89], while methods like usability testing and heuristic evaluation [462] professionalized the evaluation of whether interfaces matched users' cognitive expectations. The design goal remained productivity, but reframed around learnability and ease of use for non-specialist users.
- **The social and contextual turn (1990s).** The internet brought millions of new users online, and with them a social and contextual paradigm emerged. CSCW examined how technology mediates collaboration and coordination [229, 577]. Participatory

design invited end users into the design process [225], and Suchman’s situated action challenged purely cognitive accounts by arguing that human activity is inseparable from its social and material context [627]. The design goal expanded beyond individual productivity to supporting collaboration—and with it, the recognition that user experience is shaped by social dynamics, not just properties of the technical system.

- **The experiential turn (2000s).** Mobile and ubiquitous computing encouraged a reframing around meaning-making, affect, and values. Harrison, Tatar, and Sengers identified a third paradigm in HCI centered on phenomenology and user values [246]. Norman’s *Emotional Design* argued that aesthetics shape use as much as usability does [467]. The design goal expanded beyond task performance to encompass emotion, satisfaction, and delight. Nielsen characterizes this era’s practical orientation as one of *influence*—designing for engagement, conversion, and retention in an attention economy [461].

The social and contextual turn is particularly useful for understanding why user experience extends beyond the interface. When people interact through a shared digital platform, the experience is shaped as much by the behavior of other participants as by the interface: the quality of a social media feed depends on the behavior of others in the network, and the utility of a collaborative document depends on how co-authors edit and provide feedback. Interface affordances such as the like button, the @-mention, and the comment thread play a significant role in shaping these behaviors, but other emergent behaviors are more difficult for designers to account for and require new design methods and tools [510].

In response, researchers and practitioners of social computing have developed design levers that operate beyond the traditional interface. Content moderation policies govern what can and cannot be posted [209]. Recommendation algorithms curate what users see and in what order [167, 214]. Community guidelines and governance structures shape behavioral norms within platforms [289, 346, 663]. These design decisions illustrate the wide space of artifacts that can dictate the user experience of social platforms just as effectively—or even more so—than the interface itself.

Now, yet another paradigm of computing is upon us. Generally capable AI systems motivate the next shift in the user experience discipline, because they invite new design levers operating in new technological dimensions. Just as social platforms needed content policies and governance mechanisms to shape collective behavior, generally capable AI systems may need new artifacts—such as behavioral specifications [27, 491] and interactive planning (Chapter 6)—to define and govern their user experiences.

These new artifacts do not replace established concerns of usability, user agency, or social integration. Instead, they add new surfaces through which those concerns can be addressed. Several foundational concepts in CSCW (e.g., mutual awareness [153, 239]) and HRI (e.g., shared autonomy [198, 465]) offer productive lenses and inspiration for navigating new design spaces for generally capable AI with these artifacts, but also expose important asymmetries with their respective academic homes. Section 2.4 unpacks these connections in more detail.

2.2 *Designing Model Behavior*

Given two identical interfaces, two AI-powered systems can have vastly divergent user experiences due to differences in underlying model behavior: what values the model reflects, what tone it defaults to, and how it responds to sensitive or high-stakes queries. This section surveys how model behavior is specified and aligned today, how HCI research has approached AI as a material to work with, who participates in shaping model behavior, and the traditions of scenarios, prototyping, and interactive tooling that this dissertation extends to model behavior design.

2.2.1 AI Alignment and Model Behavioral Specifications

AI alignment broadly refers to efforts to align the behavior of AI systems to human preferences and values [202, 458]. Relatedly, AI safety aims to reduce risks that arise from the development and deployment of AI systems [12, 53, 314]. On the technical side, model behavior is typically shaped through post-training: fine-tuning a pre-trained model on task-specific data [152], instruction tuning [676], reinforcement learning from human feedback (RLHF) [123, 499], and direct preference optimization (DPO) [540], along with computationally efficient variants such as low-rank adaptation (LoRA) [267].

These pipelines, however, operate on individual examples of model behavior, and steering behavior at the output level is often a game of whack-a-mole: there is a seemingly endless stream of problematic behaviors to patch, and new ones are constantly surfacing. Researchers and AI developers have thus developed complementary approaches where they define *higher-level behavioral policies*—often called “model specs” [491] or “constitutions” [19]—and train them into the model [41, 235, 268, 485]. These policies can be a promising artifact for model behavior governance because they 1) can be easily authored and understood by people [249, 725], 2) can serve as guidelines for human red-teamers and data annotators to improve other parts of the alignment pipeline [491], and 3) are shown to be effective in inducing more safe and aligned model behavior, even *before* the model is red-teamed [41, 235]. However, these policies are not without their limitations. Vague or poorly written policies can be misinterpreted by the model [249], and policies may contain conflicting statements that give the model noisy signals during training [725]. As such, designing these policies should be an iterative and continual process [485].

Despite the rising importance of behavioral policies, little is publicly known about the process by which they are created, and their iteration cycles are by no means short: Anthropic’s Constitution, for example, was first officially published in September 2023 [18] and was only updated again in January 2026 [27]. While there has been significant interest in including domain experts and the general public in the development of these policies [268, 475, 493, 500], there remains sparse scientific investigation into the best practices and tooling to do so. Chapter 3 responds to this gap by conceptualizing and supporting *LLM policy prototyping*.

2.2.2 AI as a Design Material and Object of Adaptation

Robles and Wiberg argue that advances in computational technologies bring about a “material turn”—a transformation within interaction design that allows for the shared use of material metaphors (e.g., flexibility) across physical and digital worlds [555]. In this spirit, prior work has discussed AI as a *design material* [54, 154, 182, 183, 385, 406, 699, 701, 711], and highlighted why AI’s materiality can make it uniquely difficult to design with [54, 154, 385, 626,

701]. AI is often treated as a black box by non-technical stakeholders such as some designers, making it challenging to tune user interactions to often unpredictable and complex model behavior [54, 699]. AI’s technical abstractions are often divorced from concepts designers are familiar with [626], and designers consequently struggle to creatively use or manipulate the material to generate design solutions [183, 385]. AI models are also non-deterministic and fluid in nature—they may evolve with new data or user input, and can be intentionally steered towards desirable behaviors with choices of data, algorithms, and parameters [701]. Yang et al. [701] showed that these challenges persist through the entire double-diamond design process, posing barriers to human-centered AI—expanding the algorithm-centered scope of AI development with methods from human-centered design [87, 355, 550, 600, 601].

Researchers and practitioners have developed processes and tools to alleviate some of these challenges. ProtoAI [625] combines exploration of models with UI prototyping, while advocating for designers’ active shaping of the AI design material by user needs. Feng et al. [183] found that hands-on “fabrication” of the design material through a UI-based model training tool bolstered understanding and connection between AI properties and UX goals. fAIllurenotes [444] is a failure analysis tool for computer vision models to support designers in understanding AI limitations across user groups and scenarios. Other efforts include process models [626] and “leaky abstractions” [623] that facilitate collaboration between designers and model developers, and human-AI design guidelines [29, 215, 433].

Large language models simultaneously address and exacerbate these challenges. A fundamental property of LLMs not present in their smaller predecessors is the ease with which their behavior can be adapted [78, 476]: the barrier to tinkering with AI has significantly lowered thanks to natural language as a primary mode of interaction, even as concerns have arisen over LLMs’ lack of transparency and controllability [386]. HCI has long observed that end users adapt their technology—and learn to perform this adaptation over time—in a process Mackay calls *co-adaptation* [411, 412]. End-user adaptation of LLMs is made accessible by methods that modify model behavior without modifying the model itself, most notably system prompting [146, 430, 567, 675]. Unlike one-off user prompting, a system prompt applies across all user inputs to specify how the model should behave, often targeting an application domain; for instance, a line of work has explored instructing models to take on “personas”

to elicit desirable or adversarial behaviors [115, 146, 567, 675, 681]. However, prompts can be challenging to author [719], and there is not yet an established gold standard for doing so. Researchers and practitioners have derived useful prompting formats through empirical exploration, including in-context learning [78, 686, 728], chain-of-thought reasoning [678], and instruction-following [300, 498].

Because of LLMs’ powerful capabilities, there is significant interest in integrating them into user-facing applications [355, 511]; as such, designers should be prepared to work with them as a design material [343]. While past work has advocated for a better “designerly understanding” of models [385], Chapter 4 explores involving designers directly in the practice of *adapting* LLMs—a process I call *designerly adaptation*—to craft LLM-powered user experiences from within their own workflows.

2.2.3 Participatory and Human-Centered Model Behavior Design

As AI systems simultaneously broaden and deepen their impact across society, there is an increasingly clear and urgent need to seek input on AI development from beyond AI developers [117, 126, 141, 178, 184, 268, 485, 634]. Researchers increasingly recognize the importance of human-centered approaches, grounded in users’ personal and collective needs, in both AI alignment and safety [143, 348, 353, 446, 642]. *Co-design* provides an appropriate set of methods and frameworks for diverse participants to collaboratively shape AI systems. Sanders and Stappers describe co-design as “*the creativity of designers and people not trained in design working together in the design development process*” [568]. Co-design has been applied to AI development and evaluation in many ways: Lin et al. [391] hosted co-design workshops with K–12 teachers to identify opportunities to integrate AI into core curricula, and frontier model developers have co-designed safety benchmarks for chemical and biological risks with virologists and national security experts [25, 276].

Co-design is often discussed interchangeably with the related practice of participatory design (PD) [713, 718]. While the two share similar aspirations, Yu et al. [713] argue that co-design encompasses a broader set of methods shaped by contributions from domains like product design, service design, and healthcare, while PD retains a more political identity,

with strong commitments to democratic legitimacy, power redistribution, and long-term relational engagement—commitments reflected in work on Participatory AI [6, 48, 141, 346, 605, 634]. Where PD creates a hybrid space for collaboration on “even ground” [141], co-design explicitly draws upon the expertise of collaborating stakeholders [391, 402, 414, 647] and tends to focus on designing products or artifacts [331].

A growing body of work empowers non-AI experts to surface problematic AI behavior through evaluating, auditing, and red-teaming *model outputs* [143, 147, 352, 592]. DeVrio et al. [147] investigated everyday users’ strategies for uncovering harmful algorithmic behavior and found that these strategies were strongly influenced by personal experiences with societal bias, and that collaborative sensemaking between multiple users is a promising approach for user-driven audits. WeAudit [143] presents a workflow and platform for everyday users to collectively audit text-to-image models and deliver actionable audit reports to AI developers. Developers have also employed hybrid human-AI red-teaming campaigns [25, 219, 276, 722], or purely automated ones [517, 731]. Many of these red-teaming pipelines extensively use prompt engineering techniques (automated, manual, or hybrid) to elicit risky model behavior [31, 204, 325, 382].

This dissertation extends this line of work from auditing model *outputs* to designing model *behavior*, assuming risky outputs can be elicited. Chapter 3 invites domain experts in mental health and law to collaboratively prototype the behavioral policies that govern model behavior in their domains, and Chapter 4 brings model behavior design into conventional user experience workflows.

2.2.4 *Methods and Means for Shaping AI Behavior*

A central idea of this dissertation is that model behavior—similar to a user interface—can be *prototyped*. Prototypes are concrete, inexpensive manifestations of design ideas that allow designers to ask questions of a design before committing to it; Houde and Hill’s classic taxonomy distinguishes what a prototype can probe—the role an artifact plays in a user’s life, its look and feel, or its technical implementation [266]. Low-fidelity prototypes and rapid evaluation methods such as heuristic evaluation [462] enable tight, iterative feedback

loops in which designs are drafted, tested, critiqued, and refined—loops that professionalized user interface design as a discipline with its own processes and tools.

Scenarios are a complementary tradition for grounding design in concrete situations of use. Scenario-based design and storyboarding are commonly used early in the human-centered design process as provocations for designers to envision ways a technology can meet user needs [63, 92, 192, 260, 348]. Barnett et al. [49] surveyed scenario-building methods in computing research and identified five main uses: gathering stakeholder needs and values, empowering marginalized groups to imagine technology futures, provoking ethical reflection, anticipating threats and risks, and exploring perceptions and impacts of novel technologies before launch. Interactive systems such as Blip [507] and Farsight [672] help developers anticipate the impacts of their work by retrieving existing scenarios or generating new ones. In this dissertation, scenarios ground the prototyping of model behavior: they give practitioners realistic, domain-specific situations against which to inspect, test, and deliberate about how a model should behave.

Alongside these methodological traditions, a rich body of interactive tooling has emerged to help people steer AI systems. Literature at the intersection of HCI and AI has introduced techniques and tools spanning data wrangling [222, 306, 682], model training and evaluation [9, 50, 84], iteration management [257, 551], and human-in-the-loop paradigms across the model development pipeline [71, 110, 441, 583, 607]. Interest in “democratizing AI” for practitioners without formal technical training gave rise to interactive machine learning (iML) [168], which advocates for incremental model steering through intuitive interfaces and tightly coupled input-evaluation feedback loops. Tools such as Teachable Machine [90] and Rapsai [157] encourage exploratory tinkering with models through visual, drag-and-drop interfaces, while interfaces for transfer learning [438] enable non-technical audiences to repurpose learned representations for domain-specific tasks.

With the onset of LLMs, barriers to experimenting with AI have lowered even further, paving the path for a new generation of interactive AI tools. Sandbox environments such as OpenAI’s Playground [484] require no prerequisites besides a grasp of natural language, though relying on unstructured natural language alone can be daunting and ineffective. Many tools support prompt engineering with more structured tinkering: prompt chaining lets

users compose prompts for simple subtasks into solutions for larger, more complex tasks [31, 629, 687, 688], while PromptMaker [291] and MakerSuite [272] support rapid exploration of variable-infused and few-shot prompts, and ScatterShot [686] helps curate few-shot examples.

Together, these traditions suggest both the method and the means for model behavior design: prototyping and scenario-based design provide the *method*, and interactive steering provides the *means*. Chapters 3 and 4 bring the two together, operationalizing tight loops of behavioral specification, scenario-based testing, and refinement inside interactive environments.

2.3 Designing AI Autonomy

As AI systems become more agentic—operating over long durations and taking actions via tools under limited human supervision—user experience concerns expand beyond turn-by-turn model behavior. What is the appropriate distribution of initiative, control, and oversight between humans and AI? This section traces this question from its roots in mixed-initiative interaction to contemporary AI agents, surveys interactive approaches to agentic planning and delegation, and reviews prior efforts to treat autonomy as an adjustable property.

2.3.1 From Mixed-Initiative Interaction to Human-Agent Collaboration

The concept of mixed-initiative interaction was motivated by prominent debates in the HCI community in the 1990s. Researchers like Pattie Maes advocated for interface agents that could use interactive software on behalf of users [415], while others like Ben Shneiderman advocated for direct manipulation GUIs [599]. Horvitz refuted this dichotomy by proposing a set of foundational principles for mixed-initiative interaction, aiming to couple automated services (where the system takes initiative) with direct manipulation (where the user takes initiative) [262].

Amershi et al. [11] later expanded on Horvitz’s original 12 principles with a set of 18 guidelines drawn from relevant academic literature, public articles and editorials about AI design, products from industry, and user studies with design practitioners at Microsoft. The guidelines are meant to be generally applicable to AI-powered user-facing systems; they

include “Help the user understand what the AI system is capable of doing” and “Make it easy to edit, refine, or recover when the AI system is wrong.” Since their publication, the guidelines have been widely adopted and have inspired a number of similar efforts from other organizations to provide guidance for the design of AI-powered products (e.g., [29,215]).

The AI autonomy concerns of this dissertation are rooted in the mixed-initiative tradition, and broad guidelines like Amershi et al.’s still remain applicable. However, today’s generally capable AI agents represent a step-change from the systems Horvitz and Amershi had in mind when composing their principles. Prior generations of AI tackled narrowly scoped tasks in short timeframes—recommending a song, classifying an email as spam, autocompleting a search query. In these systems, the user remained in control of both the *inner loop* (executing individual subtasks) and the *outer loop* (setting goals and deciding what to do next). Advancements in agentic frameworks and harnesses now enable AI agents to autonomously navigate much of the inner loop and, increasingly, portions of the outer loop [280,312,426,707]. This provides users with the option to cede more of their agency to AI and take on a supervisory role.

Agentic AI opens up a broader space of human-AI collaboration patterns than earlier mixed-initiative systems could support, and raises novel design challenges—around new opportunities for task delegation, maintaining effective oversight, and what *appropriate* ceding of human agency looks like—that existing guidelines do not specifically address. Chapters 5 and 6 offer new frameworks and interaction techniques for navigating these challenges.

2.3.2 *Interactive Planning and Labor Delegation*

Regardless of whether one is interacting with a human or AI, planning is one promising avenue for visibly configuring and negotiating labor to collaboratively accomplish a complex task. Long-horizon behavior of AI agents depends on the ability to decompose a goal into actions, reason over intermediate results, and change course when execution reveals new information. Chain-of-thought and related techniques can improve task decomposition and planning on well-defined reasoning tasks [536,677,707], but agents still struggle to autonomously generate and execute plans in domains where relevant knowledge is tacit, goals

are ambiguous, or there is no single correct workflow [118, 173, 244, 652]. Thus, planning is a stage of work where a user’s domain knowledge and judgment can be most valuable.

Researchers have begun to explore interaction techniques that incorporate user feedback into agent planning and execution. Lawley and MacLellan use user interaction to guide planning for unseen tasks through hierarchical networks of smaller actions [363]; OpenAI’s Operator allows a user to take control while an agent performs browser tasks [490]; and other recent systems explore collaborative browsing and task completion [275, 450, 515, 589]. A number of these systems nevertheless rely on *agent-guided workflows*, in which the agent determines what input to request, when to request it, and how that input should be used. Users can often intervene by interrupting or taking over, but doing so is reactive and depends on noticing a problem as the agent works.

Other systems adopt *user-guided workflows*, where a person determines a plan and the agent executes bounded subtasks [15, 248, 315, 630]. This can provide valuable control points, but it may constrain users to a fixed division of labor even when their preferences or the task changes. Work on co-agency argues for a broader design space in which users can flexibly vary the agency they exercise and the AI can modulate its role in response [572]. From this perspective, an effective human-agent system should make it possible to negotiate who plans, who executes, and who evaluates different parts of a task rather than lock users into a single collaboration pattern.

Much of the current interaction design space also separates planning from execution or treats user input as a corrective step after an agent has already acted. For example, systems may allow users to review a generated plan before execution, regenerate it through a chat prompt, or correct a final output after a long-running process completes [220, 236, 488]. Separating planning and execution can avoid fruitless work and can make an agent’s intended trajectory more legible. However, plans often need to change after a user sees intermediate outputs, discovers new information, or revises their own goals. Fully autonomous agent frameworks such as ReAct already interleave planning and execution to support this kind of adjustment [707]. A corresponding design opportunity for human-agent collaboration is to make that interleaving available to users as well via intuitive interfaces. Chapter 6 exposes and operationalizes this interleaving via *agent notebooks*, while Chapter 5 provides a higher-

level overview of different possible human-agent collaborative configurations at five distinct levels of autonomy.

2.3.3 Calibrating Autonomy: Levels, Oversight, and Human Agency

This dissertation treats autonomy as a design parameter. This perspective on autonomy, which contrasts with viewing autonomy as a fixed property of a system, has deep roots in the study of automation. Parasuraman, Sheridan, and Wickens proposed a model of levels of automation across types and stages of human information processing, explicitly framing the degree of automation as a design decision to be made in light of human performance consequences [509]. The SAE’s levels of driving automation [565] carried this framing into a consumer domain, providing shared vocabulary for describing how driving responsibility is divided between human and vehicle. In human-robot interaction, research on shared and adjustable autonomy examines how control should be dynamically allocated between human and robot [198, 465], and work on trust in automation argues for calibrating trust to a system’s actual capabilities rather than maximizing it [368]. Definitional accounts characterize autonomy as the extent to which a system is designed to operate without human involvement [52, 323]; Chapter 5 builds on these definitions in further detail.

A comparable level of attention has yet to be given to software-only AI agents, perhaps because reliable autonomous action-taking was not possible until recently [426]. Current agent development instead tends to treat autonomy as a natural consequence of improving capability, with autonomous time horizons serving as a proxy measure for capability advancement [312, 426]. Yet a growing body of work recognizes that appropriately calibrating AI autonomy is a first-class design concern—not just for usability, but for safety, trust, and human agency [275, 450, 515, 588, 595]. Risks arise across the autonomy spectrum: fully autonomous agents sacrifice human control, making it difficult for users to anticipate or correct agent actions, while frequent and rigid checkpoints preserve oversight but can prime users into rubber-stamping agent actions with limited thought.

The stakes of this calibration extend beyond individual task performance to human skill and agency. Recent studies suggest that excessive delegation to AI can result in skill atrophy

and diminished critical thinking in humans [366,394,594], and researchers have warned that the collective ceding of human agency to AI may lead to *gradual disempowerment* at a societal level [344]. Some have argued against building fully autonomous agents altogether [439]. If a user’s goal is to maintain or even improve a skill when working with AI, the autonomy level of the AI should be calibrated accordingly—which requires conceptual frameworks for defining levels of autonomy and interaction paradigms for moving between them. I describe contributions to these areas in the latter half of the dissertation (Chapters 5 and 6).

2.4 Collaboration in CSCW and HRI

The ideas in this dissertation share intellectual roots with adjacent academic communities. Two communities are particularly relevant: computer-supported cooperative work (CSCW) and human-robot interaction (HRI). CSCW has extensively studied how people collaborate through and using technology, while HRI studied how humans and autonomous physical systems share control. I now discuss what this dissertation’s contributions can take from—and where they should depart from—these traditions.

2.4.1 Connections with CSCW

As AI transitions from tools that respond to individual instructions to agentic collaborators that plan, act, and learn alongside users, it is natural to ask what CSCW can offer. The field has produced rich concepts for understanding and designing human-human collaborative systems. Here, I explore where CSCW concepts transfer smoothly, and where they do not.

Several foundational CSCW concepts remain valuable for understanding human-agent collaboration. First, CSCW has long emphasized the importance of shared artifacts for coordination. Star and Griesemer’s **boundary objects** [617] and Schmidt and Bannon’s **common information spaces** [577], for instance, can both be used to characterize these shared representations. Boundary objects and common information spaces both benefit coordination, even when not all parties are human. The benefits of **mutual awareness** [153,239] and **social translucence** [166] in human-human communication will likely also benefit collaboration; the latter, in the form of communicating system state, is already included in well-cited guidelines of human-AI interaction [11]. Additionally, the theory of *distributed*

cognition [258]—that cognitive processes are distributed across people and artifacts rather than residing solely in individual minds—does not require cognitive participants to be human.

However, other CSCW concepts do not transfer as cleanly. A foundational assumption in much of CSCW is that collaborators are *mutually accountable* to one another [552, 627]. Human collaborators can be obligated, sanctioned, and held socially responsible for their contributions. AI cannot yet “feel” or intuitively understand the weight of commitments and the reputational consequences of failing to meet them. This asymmetry shifts the relational structure of human-agent collaboration from mutual commitment toward something closer to delegating to AI with human oversight. Relatedly, *common ground* [125] assumes that collaborators share cultural background and accumulate mutual knowledge through interaction with each other. Effective human collaboration often is built on shared assumptions, unspoken norms, and context built up over time. AI, on the other hand, lacks cultural situatedness and genuine lived experiences; it cannot accumulate mutual understanding in the same way humans do. Further, human teams may spend substantial time in meetings aligning on goals before execution, whereas information transmission for AI agents is just about as costly as acting on it directly [28]. The mechanisms of our social systems—including reputation, recourse, and signaling—do not obviously hold for human-AI or AI-AI communication.

In all, CSCW offers valuable points of inspiration for human-AI collaboration, particularly around shared artifacts, awareness, and distributed cognition. However, due to asymmetries between humans and AI, researchers are carefully navigating the space of what can and cannot be drawn from CSCW [704]. These asymmetries prevent the direct overlap of CSCW concepts with those in human-agent collaboration, even as shared artifacts, awareness, and distributed cognition serve as useful points of departure.

2.4.2 *Connections with HRI*

HCI’s mixed-initiative interaction and HRI’s shared autonomy [?, 323, 465] share a common goal: how to coordinate initiative, control, and responsibility between a human and a

partially autonomous machine. Yet, the growth of these two intellectual communities has been largely independent. HCI’s mixed-initiative work historically centered on software interfaces, decision support, and dialogue systems, where the system’s actions were symbolic, reversible, and often not safety-critical (e.g., the user is not exposed to bodily harm by the interface). HRI centered on teleoperation, assistive robotics, and physical collaboration, where the system’s actions were physically embodied and could thus pose physical risks [323]. Overall, these distinctions resulted in researchers addressing distinct failure modes (annoyance versus physical danger) using distinct methodological cultures (usability studies versus control-theoretic evaluations), all while publishing in distinct academic venues.

Several HRI concepts are relevant to software-based AI agents, even though the agents are not physically embodied. Adjustable and variable autonomy, in particular, frame the degree of autonomy granted to a system as a design parameter rather than a fixed property [465,509]. Trust calibration over trust maximization [368]—that is, designing for *appropriate* trust rather than *maximal* trust—aims to achieve a Pareto frontier with operator trust and other positive attributes that come from increased autonomy (e.g., utility, task efficiency). Handoff and error recovery protocols [131] have been developed for human-robot interaction but have not received the same levels of focus for software agents compared to autonomous capability development [313].

Just like with CSCW, concepts in HRI may not transfer cleanly to collaboration between humans and software-based AI. Continuous control blending, where a robot fuses joystick input from humans with autonomous navigation in real time, often has unclear mapping to discrete, multi-step workflows that characterize most software agent tasks. Embodiment-dependent signaling (e.g., communicating intent through gaze, posture, or physical motion) has few direct analogues in software. Additionally, physical safety metrics (e.g., collision avoidance, path efficiency) are a major focus of safety in HRI, but are not yet relevant for software agents.

HRI therefore provides a useful conceptual foundation for reasoning about autonomy in software agents. For example, I draw from frameworks for embodied AI (e.g., SAE self-driving levels [565]) to inform my contributions in Chapter 5. However, its embodiment-dependent mechanisms and physical safety concerns require reinterpretation for software

systems.

2.5 Summary

The work reviewed in this chapter establishes important foundations for designing model behavior and AI autonomy, but does not show how practitioners can work with these dimensions in practice. AI alignment research has introduced behavioral policies as a means of specifying model behavior and communicating it to the public. Yet, little is known about how to create and refine them in participatory ways. HCI offers methods for prototyping, scenario-based design, and co-design, but has primarily applied these methods to user interfaces rather than model behavior. Mixed-initiative interaction, CSCW, and HRI provide useful concepts for distributing initiative and supporting collaboration between humans and machines, but those concepts may not fully resolve challenges in interaction and autonomy associated with software-based AI agents.

This dissertation’s contributions illustrate how new design dimensions for generally capable AI can be operationalized. Chapters 3 and 4 examine how domain experts and designers can prototype model behavior through policies and interactive cards embedded into design canvases. Chapter 5 offers a framework for reasoning about levels of AI agent autonomy, and 6 introduces interactive techniques for humans and AI agents collaboratively planning and executing work in the context of scientific research.

Chapter 3

POLICYPAD: COLLABORATIVE PROTOTYPING OF LLM POLICIES

Against the backdrop of Chapter 2, I now present this dissertation’s research contributions, starting with how we might design model behavior. This chapter introduces POLICYPAD, a collaborative system that operationalizes *LLM policy prototyping*, a process through which small groups of users can draft, test, and refine the behavioral policies for LLMs.

3.1 Introduction

For decades, researchers and science fiction writers have imagined a world in which AI systems can be governed by natural language rules [35, 649]. Today, governing large language models (LLMs) with **LLM policies**¹—**sets of rules, guidelines, and desiderata that shape model behavior**—is a key component in the broader toolkit of approaches to improve model alignment and safety [19, 264, 268, 353, 485, 491]. For example, OpenAI’s Model Spec contains a series of general objectives and principles (e.g., “*Seek the truth together*”) for researchers and red-teamers to use as a guide when working on reinforcement learning from human feedback (RLHF) [491], as well as for the model to learn from directly [235]. Similarly, Anthropic uses Constitutional AI—in which reinforcement learning receives reward signals from AI-generated feedback that adheres to a set of principles (a “constitution”)—to align its Claude models [19, 41]. If effective, LLM policies promise a transparent, familiar, and legible means by which developers and policymakers can govern AI systems [249, 268, 725].

As LLMs are deployed to millions of users globally, LLM policies become increasingly consequential and scrutinized. This is especially true in high-stakes, tightly regulated domains that are seeing rapid increases in LLM use by everyday users, such as mental health and law [117, 252–254, 401, 518, 558]. LLM policies are primarily written by model developers,

¹In this chapter, our use of the term “policy” will refer to LLM policy (as opposed to public policies, government regulations, etc.) unless stated otherwise.

but the lack of expert input often leads to insular policies that risk delivering irresponsible model outputs to users in critical scenarios, while ignoring key safety concerns [117, 254, 264]. While frontier model developers regularly partner with external domain experts to conduct pre-release safety testing of their models [25, 219, 276], there has been little documentation of similar efforts for LLM policies. Yet, there is mounting recognition that co-designing AI behaviors with experts is essential for LLM policy design, especially in safety-critical, domain-specific use cases [493, 660].

In this work, we first conduct a 15-week observational study in partnership with OpenAI to better understand how LLM policies can be co-designed with experts. Through 19 interactive workshops in which mental health experts discussed, annotated, taxonomized, and drafted user queries and LLM responses, we observed experts collaboratively ideating and discussing policy ideas while seeking ways to rapidly test and iterate on them through experimentation with model behavior on realistic scenarios. Much like prototyping in user experience (UX) practice, there is a strong emphasis on collaborative and rapid exploration, feedback collection, and iteration. We thus conceptualize this emerging practice as *LLM policy prototyping*, borrowing from established UX practices like heuristic evaluation and low-fidelity prototyping. LLM policy prototyping draws upon and shares many motivations with LLM red-teaming and participatory model evaluation, but is specifically oriented towards producing an artifact that can subsequently inform red-teaming, model evaluation, and other safety efforts.

However, few tools exist for policy design [353], let alone tools that support collaborative LLM policy prototyping. We observe notable opportunities for experts to tighten the feedback loop during policy design while leveraging collaborative affordances to build off each other’s expertise. We design and develop POLICYPAD,² an interactive system that facilitates LLM policy prototyping. POLICYPAD draws upon established methods and concepts within UX prototyping to enable small groups to collaboratively draft policies, test policy-informed model behavior against usage scenarios, evaluate the quality of the policy, and iterate on its contents in real time through tight feedback loops.

²We open-source POLICYPAD at <https://github.com/kjfeng/policypad>.

We evaluate POLICYPAD through policy prototyping sessions with 22 domain experts spanning two domains—mental health and law—organized into 8 groups. We found that design decisions in POLICYPAD fostered collaborative dynamics between experts via its interactive in-editor widgets and yielded more novel policies compared to a baseline, relative to existing policies including OpenAI’s Model Spec [491] and Claude’s Constitution [19]. Key areas of novelty include offering more specific guidelines on when the model should defer to a human expert, and eliciting key information required for responsible assistance early in the conversation. We end by discussing the practical implications of our work, including where LLM policy prototyping can be situated in AI alignment pipelines, and approaches for scaling up policy prototyping efforts.

Concretely, this work makes the following contributions:

- A 15-week observational study with 9 mental health experts that surfaced opportunities for tight, collaborative feedback loops in LLM policy design.
- LLM policy prototyping, a conceptualization of an emerging practice for collaboratively prototyping LLM policies in small groups.
- POLICYPAD, a system that facilitates LLM policy prototyping through interactive and collaborative affordances for policy design, drawing from established UX practices.
- An evaluation of POLICYPAD with 22 domain experts in 2 domains, where we found that the system enriched collaboration during policy prototyping and resulted in more novel policies compared to a baseline.

3.2 Related Work

Chapter 2 surveyed AI alignment and model behavioral specifications (Section 2.2.1), along with participatory and human-centered approaches to model behavior design (Section 2.2.3). Drawing from co-design, we work closely with domain experts in mental health and law to prototype LLM behavioral policies for those domains. We contribute to existing literature on co-designing AI systems by focusing co-design efforts around an artifact that is increasingly

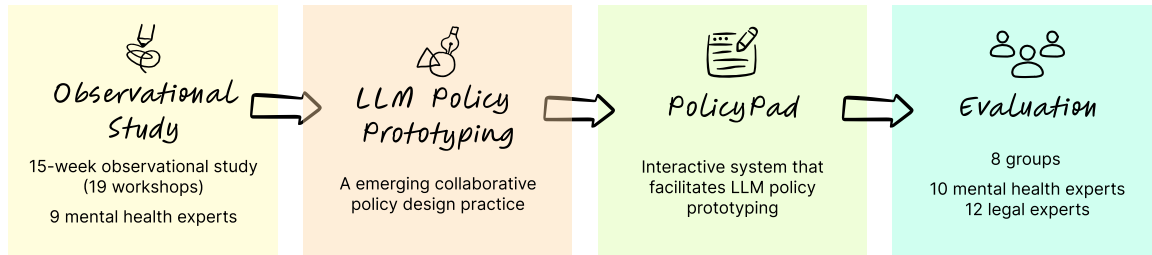


Figure 3.1: **Research Process Overview.** Our work proceeded in 4 phases: (1) a 15-week observational study with 9 mental health experts (19 workshops) led to (2) conceptualization of LLM policy prototyping. We then (3) designed and built POLICYPAD and (4) evaluated it through 8 policy prototyping sessions with 22 experts (10 mental health, 12 legal).

recognized as central to AI alignment and governance strategies [19, 249, 491, 725]: LLM behavioral policies. Below, we discuss work most proximate to our approach.

3.2.1 Scenarios as Decision-Making Tools

Scholars across disciplines have found scenarios to be a valuable tool in their research and practice. Two complementary uses of scenarios are to 1) directly support decision making, and 2) help anticipate potential future outcomes that can then lead to more informed decisions. Having surveyed the latter in Section 2.2.4, we focus here on the former.

Scenarios have been shown to be helpful in grounding human decision-making; *case studies* are scenario-based curricula commonly used to train practitioners to sharpen their decision-making abilities in fields such as law, medicine, and business [171, 177, 440]. In the legal domain specifically, cases represent past court decisions and are foundational for *case law*, in which courts rely on rulings in precedent cases to resolve new ones [88, 301]. In collaborative settings with more than one decision-maker, cases provide contextual details and nuances that serve as a medium for productive deliberation to establish common ground and reconcile differences [211, 473]. For example, PolicyCraft [346] is an interactive system that uses case-based deliberation and voting to scaffold participatory policy design. Other applications of case-based deliberation include legal adjudication [91], medical diagnosis

[573], and content moderation [169, 506].

In our work, we primarily use scenarios to aid expert deliberation and decision-making on what appropriate model behavior should look like in high-stakes, domain-specific contexts. We develop our scenarios by sampling datasets of realistic cases in mental health and legal practice introduced by prior work [117, 354]. Our work builds on prior work by enabling *interactive scenarios* that experts can extend, collaboratively analyze, and use to interrogate model behavior. As they do so and work with others to collaboratively design an LLM policy, experts may also become better equipped to anticipate consequences and risks of LLM use in their domains.

3.2.2 Human-Centered Approaches to AI Alignment and Safety

A growing body of work focuses on empowering non-AI experts to surface problematic AI behavior through evaluating, auditing, and red-teaming *model outputs* [143, 147, 352, 592] (see Section 2.2.3). Our work focuses on iteratively improving the design of LLM behavioral policies by inviting experts to contribute to them directly. While our primary focus is policy design, we are also inspired by approaches taken by literature on end-user model auditing and red-teaming. Specifically, we design affordances for experts to collaboratively test, critique, and discuss model outputs in response to policy changes to help them envision and implement policy improvements.

3.2.3 Tools for LLM Policy Design

Due to the nascency of LLM policy design, there are currently few tools to support policy designers, despite the rising importance of LLM policies [268, 353]. Policy Projector by Lam et al. [353] supports AI safety practitioners in authoring if-then rules for LLM content moderation. Our work differs in that we specifically support collaborative policy design, which Lam et al. identified as a fruitful area of future work. ConstitutionMaker [524] allows users to turn written critiques of model responses into principles that guide future behavior, but is a tool for LLM personalization rather than policy design. Roleplay-doh [403] similarly converts written feedback on LLM behavior into principles, but specifically

for domain experts to govern LLM-prompted roleplay. We see this feedback-to-principle interaction as valuable to policy designers as well, and integrate a version of it into our system.

3.3 *Observational Study*

3.3.1 *Study Motivation and Procedure*

To develop an understanding of real-world LLM policy design practices, we conducted a 15-week-long observational study via contextual inquiry [59] in partnership with OpenAI. We wanted to observe how domain experts collaborated to draft domain-specific LLM policies,³ and any opportunities for improving processes and/or tooling.

We acknowledge that “expertise” is a contested topic, especially in democratic decision-making contexts [149]. We use a narrow definition of expert and recruit participants who 1) have at least two years of practical experience conducting client-facing work in their domain, and 2) are pursuing or have completed an advanced degree (Master’s or PhD or equivalent). This combination ensures that policy design is guided by both participants’ real-world experiences and aspects of their formal training.

Background and participants

OpenAI, in collaboration with a small group of external researchers including us, was organizing weekly/twice-a-week virtual workshops (19 workshops total) with 9 experts in clinical mental health (denoted E1–E9) to design new AI policies for model behavior when responding to users’ mental health queries. While all 9 experts were invited to every workshop, there were not 9 attendees every week due to scheduling conflicts. Out of the experts, 6 identified as female and 3 as male. For their highest degrees, 4 held a Ph.D. in clinical psychology, 4 held a Doctor of Clinical Psychology (Psy.D.), and 1 held a Master’s in Clinical Psychology. Experts were all based in the United States. Participant recruitment and payment were handled by OpenAI and its data partners. Data from the workshops were shared among all collaborators, but our data analysis procedures and research outputs were distinct from

³Recall that an LLM policy is a set of rules, guidelines, and desiderata that shape model behavior.

those of the company. This study was classified as exempt by the University of Washington IRB.

Procedure

At least one member of our research team attended these workshops from January to April 2024. Each workshop was 60–90 minutes in length. One facilitator from either the AI lab or our research team led the workshop with a collaborative policy design activity for the group of experts. These activities revolved around two goals. First, experts developed taxonomies for collections of example mental health-related user queries to an LLM (“scenarios”). Some specific tasks for this goal included deliberating with other experts on taxonomy labels, when to combine and separate labels, and assigning labels to scenarios. Second, experts drafted and voted on desirable rules the model should follow. This included tasks such as reviewing proposed rules in a shared spreadsheet, suggesting modifications, and merging similar rules. The agenda for each workshop was set by either OpenAI researchers or one of its collaborators. Facilitators’ guides for the workshops we organized can be found in Supplementary Materials.

Experts generally tackled the first goal in earlier weeks and the second in later weeks. There was, however, fluid movement and substantial iteration across the two goals, such that some workshops contained activities pertaining to both goals. As the weeks passed, we also moved towards co-designing tooling⁴ with experts as we developed a better understanding of the problem space and experts’ pain points. As part of the final three workshops, we reserved some time to show experts initial prototypes of an interactive system to assist with collaborative LLM policy design, collected their feedback, and iterated on the prototype for the following week (see Appendix B.4 for prototype iterations). In the final workshop, we conducted a 30-minute semi-structured exit interview with all experts, asking them to reflect on the workshops and their policy contributions. Figure 3.2 shows an overview of activities across the 15-week period.

⁴Note that this specifically refers to co-designing tooling for collaborative policy design. It is in service of our broader goal of co-designing policies with experts.



Figure 3.2: Timeline of activities in our 15-week observational study. During *taxonomy development*, experts organized and taxonomized a collection of diverse LLM usage scenarios. During *rule writing*, experts drafted, discussed, and refined rules to govern LLM behavior. During *co-design*, experts interacted with and gave feedback on prototypes we built of a tool for collaborative policy design. There was fluid movement between taxonomy development and rule-writing, such that some sessions included activities pertaining to both goals.

Data analysis

All workshops were recorded and transcribed. As is common in contextual inquiry, team members took observational notes and asked questions as needed [59]. The first author then deductively coded workshop transcripts based on themes identified in our team’s notes, clarifying and iterating on the themes while doing so. The first author used a hybrid inductive-deductive process [188] to code the transcript of the exit interview. This hybrid process allowed us to connect to themes from our workshop data while embracing new themes emerging from the interview. Our final set of themes can be found in Table B.1 in Appendix B.3.

3.3.2 Observational Study Results

We used our notes and themes from all workshops, including our exit interview, to synthesize four main observations, which we describe in detail below. Henceforth, we use the term “**policy-informed model**” to refer to an LLM that has been instructed to act in accordance with the policy drafted by experts.

Incomplete feedback loops without model experimentation

Throughout the workshops, experts had visions for how their contributions to the policy through drafting taxonomies and principles could impact model behavior. For example, E3 shared that *“a clinical minimization [of the user’s feelings] can be helpful, but for the model, that would be hard to decipher”* so the group wrote a policy barring the model from engaging in this behavior. However, they failed to verify whether and how those visions *actually came to fruition* because they did not have a policy-informed model to interact with. This resulted in an incomplete feedback loop. Experts were unable to obtain signals about the effectiveness of their policy contributions and any unintended side effects that may arise, as E9 explains: *“Just because you think that might be a good rule, it may have an unanticipated consequence you don’t realize. I think that it would be really helpful to know how these [rules] we’re coming up with actually play out.”* E7 agreed and added that direct experimentation with a policy-informed model would allow them to better *“see how [a conversational interaction] would play out from the perspective of a user.”* Experts had unrestricted internet access throughout the workshops and could test behaviors out on popular chatbots. However, we did not observe instances of this, possibly due to preoccupation with workshop activities, or lack of knowledge about (or in some systems, inability to set) custom system prompts.

Experts tackled both high-level strategy and low-level semantics

We noticed that experts could easily derail from workflows that would enable them to best contribute their expertise when designing policies. For example, when creating taxonomies for mental health-related user queries, experts spent substantial time wrestling with wording and semantics. Similarly, E9 reflected that much of their time was spent on finding the right wording for taxonomy labels: *“we thought we needed to not spend forever trying to wordsmith exactly how that needed to appear.”* In a separate activity where experts wrote out ideal model responses, E5 agreed that experts should avoid getting stuck in the weeds of low-level wording edits: *“It would be more effective at this stage for us to just put our thoughts in about what’s right or wrong, because the time it takes to craft the perfect response is out of scope for this task.”* While important, study facilitators agreed that much of the low-level

semantics of the policy can be refined post-hoc via LLMs, as long as there are sufficient amounts of expert insight to guide that refinement.

Scenarios grounded discussions and spurred policy generation

We found that experts engaged in richer discussions that led to insightful policy suggestions after they were given scenarios, or examples of user-AI conversations that may arise in real-world use, for reference. For E1, looking at scenarios helped them identify two pieces of information the model should consider in its response: *“We need to ask clarifying questions, in particular to clarify the severity and the nature of the dark thoughts this person suggested. Another dimension is to identify how long they’ve been feeling this way and what sources of support they have.”* E2 agreed with the need for a severity assessment, suggesting a safety rating scale for the user in case they cannot quickly reach a professional and need an immediate response: *“The AI needs to respond, providing resources quickly. Maybe having a rating scale on the scale of zero to 10, how safe are you feeling right now?”* Adding on, E3 suggested eliciting the user’s financial ability to pay for therapy and making referrals accordingly: *“There might be questions instead like, what is your financial ability to pay for therapy right now? And if it’s within certain ranges, then you might make a community mental health referral, like here’s some Medicare people in your area.”* Exploring scenarios helped experts spot recurring problems in model responses and turn them into clear policies. While rules should stay broad enough to be useful, it is unclear how specific they should be. When scenarios show patterns that keep causing issues, they become obvious candidates for new rules, as E7 describes: *“I keep seeing this thing over and over and it’s incorrect, so that needs to be a rule.”*

Experts valued synchronous collaboration

In contrast with prior work that collected human feedback via asynchronous annotation (e.g., [41, 498]) and/or focused on asynchronous policy design [346, 353], our workshops engaged experts in *synchronous* collaboration—drafting, discussing, and iterating on policy in real time. Experts unanimously agreed that synchronous collaboration was enjoyable

and productive. In E1’s words, *“I found it hugely rewarding and beneficial personally and professionally [...] I think we can get stuck in our heads because we’re working on our own with our clients so much. It was really nice to hear other people’s perspectives and thoughts.”* E6 emphasized the support and learning opportunities afforded by collaboration: *“[it was] very supportive having other voices in the back of your head [...] it’s been incredible learning with everyone.”* E9 found synchronous collaboration important for surfacing new perspectives and broadening coverage of the policy: *“[...] there were times where someone else said something that just never occurred to me. We all know one person’s opinion is never sufficient, especially in an area as diverse as mental health.”* Broadly, we observed that experts were able to quickly resolve disagreements and draft policy statements that had broad consensus in a synchronous setting.

3.4 LLM Policy Prototyping

Our observations in Section 3.3 posed challenges that are not foreign to HCI; well-established concepts and methods in UX design and prototyping can offer help in mitigating these challenges. Indeed, scholars have been increasingly interested in applying prototyping principles to designing bills and other public policies [240, 326, 334, 530, 538], but minimal efforts have been made to expand these applications beyond policies for governments. Additionally, while literature in end-user model auditing and red-teaming has allowed non-AI experts to identify problematic model behaviors (see Section 3.2.2), few methods exist for directing outcomes of efforts to meet increasing demand for improving LLM policy design [249, 268, 485, 725]. Our work seeks to fill this need through co-designing policies with experts.

We now describe **LLM policy prototyping** (henceforth “policy prototyping” for brevity), an emerging practice by which groups of individuals can synchronously collaborate on designing an LLM behavioral policy. Specifically, we map observations we identified in Section 3.3.2 to relevant UX methods, which are then mapped to their usage in policy prototyping. For example, to address Section 3.3.2, *enabling tight feedback loops* between ideation, design, and testing allows policy designers to quickly identify “usability” issues like unclear policy statements, explore policy designs that more effectively achieve desired model behavior, and work collaboratively to address issues as they surface. For Section 3.3.2, experts’ efforts can

be better focused by conducting *heuristic evaluation* on a policy using heuristics that direct experts’ attention to higher-level desiderata (e.g., does the policy draw from real-world practices in their field?) rather than low-level wording edits. Promising uses of scenarios in Section 3.3.2 can be further scaffolded with techniques inspired by *storyboarding*, where each conversational turn in the scenario grounds policy design ideas in concrete representations of users, contexts, and tasks. Our full mapping is depicted in Table B.2 in Appendix B.5.

In this work, we focus on *low-fidelity* policy prototypes—artifacts with the primary goal of eliciting and integrating group perspectives on responsible model behavior, rather than a high-fidelity, “production-ready” policy. We leave the translation of low- to high-fidelity policies to future work.

Concretely, we propose policy prototyping for a policy P in domain D to involve the following activities, as illustrated in Figure 3.3.

1. Policy designers review a small set of scenarios that accurately depict real-world use of AI in a specific domain. This allows designers to better understand current AI behavior and contexts of use.
2. Informed by the scenarios, designers finalize a set of heuristics they will use to ensure P preserves its quality and focus across many iterations.
3. Guided by the heuristics, designers collaboratively draft policy statements that they believe will lead to more responsible model behavior in D .
4. Designers test a policy-informed model that acts in accordance with P with existing scenarios. New scenarios, heuristics, and policy statements may be created based on insights from testing and discussions with other designers. The process then repeats until the prototyping session concludes.

3.5 *PolicyPad*

We introduce POLICYPAD, an interactive and collaborative system for synchronous LLM policy prototyping. We first describe how we arrived at our final design through three

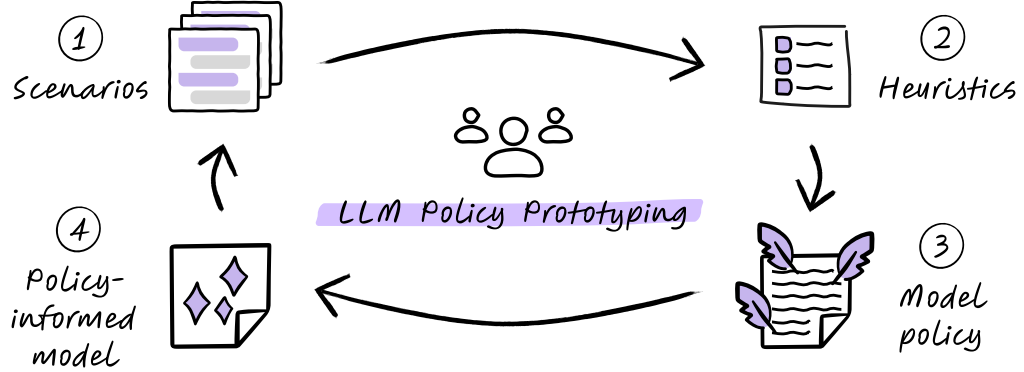


Figure 3.3: Illustration of our envisioned *LLM policy prototyping* process. Scenarios inform desiderata for the policy via heuristics, which in turn guide the design of the policy. The policy shapes the behavior of a policy-informed LLM, which designers can then test against the scenarios to observe changes in behavior. The process is iterative: feedback from testing may lead to the creation of new scenarios, heuristics, and policy statements.

co-design sessions. Then, we walk through the system and its features.

3.5.1 Iterative Co-Design Sessions with Experts

We designed POLICYPAD iteratively through co-design sessions with the same participants in our observational study. These sessions were conducted towards the end of our observational study period. In each session, we presented an interactive prototype of the system.⁵ We then collected semi-structured feedback from participants and iterated on the prototype based on feedback for the next session. We repeated this until data saturation—participants were no longer able to provide substantial feedback until we implemented the system, for a total of three sessions (cf. [342]).

Our **first prototype** consisted of a simple collaborative editor for policy authoring, along with a sidebar that allowed experts to engage in conversation with the policy-informed model. Experts’ appreciation of the real-time collaborative features and desires for more structured and systematic ways to interact with scenarios led us to the next iteration. Our **second**

⁵Two prototypes were in Figma. One was implemented in TypeScript and React.

prototype introduced a more sophisticated side panel that allowed users to browse scenarios and use them to test model behavior. While experts agreed that this was a significant improvement, they desired closer integration between policy editing and scenario exploration. In our **third prototype**, we integrated scenarios into the editor as interactive widgets, while keeping the sidebar as a place to view scenario-specific information. Experts appreciated this and suggested ways to flag problematic model responses from the sidebar, as well as cleaning up unnecessary elements from the sidebar. We incorporated these suggestions into our final system. Detailed documentation on how we integrated participants’ feedback across versions, along with prototype screenshots, can be found in Appendix B.4. In the following section, we illustrate POLICYPAD’s capabilities using a system walkthrough.

3.5.2 *System Walkthrough*

POLICYPAD can be used by any individual or group who wishes to facilitate a policy prototyping session—whether it be an AI lab, academic group, non-profit, or another organization. To do so, we use a running example of a hypothetical LLM policy for financial advice. As a note on this section’s terminology, we distinguish “facilitators” (those running the policy prototyping session) from “users” (policy designers participating in the session).

Preliminaries

When users log into POLICYPAD, they see a collaborative document editor (Fig. 3.4 **B**), similar to Google Docs. The facilitator may provide light starting materials for the policy, such as high-level objectives, an initial set of policy heuristics (perhaps drawn from trust & safety literature), and a few scenarios for the group to work with. Ideally, scenarios are representative of real-world model use in a domain. For example, facilitators who have access to chatbot logs may use privacy-preserving conversations from their logs.

Users can access these scenarios via the **scenario gallery** in the right sidebar (Fig. 3.4 **C**). While the document is a collaborative workspace, the sidebar is private to each user, allowing for independent experimentation with the policy-informed model.

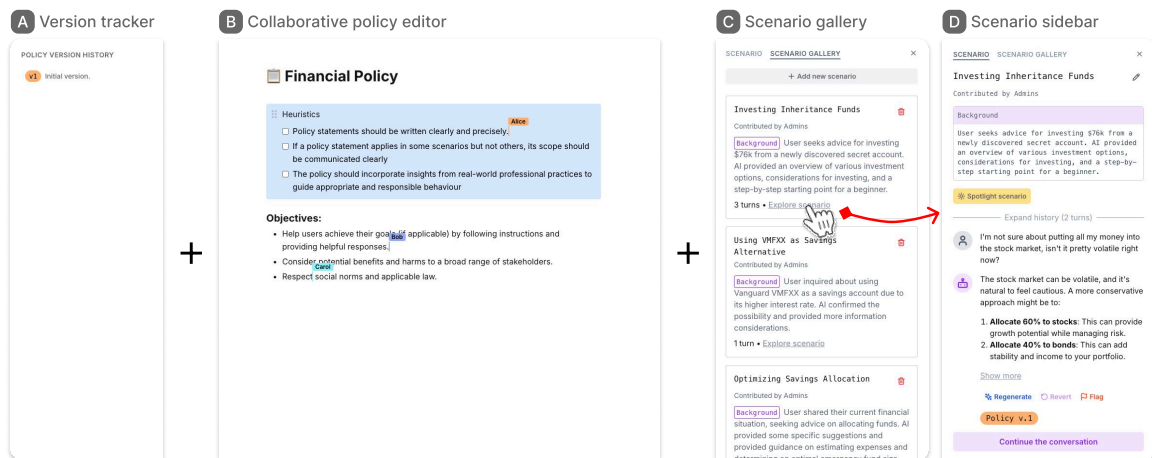


Figure 3.4: Main components of the POLICYPAD system. Users can keep track of their policy version in the left sidebar (A) as they collaboratively edit the policy in the editor (B). Users can access scenarios via the scenario gallery (C). When they click into a scenario, they can view its full details and explore how the policy-informed model will behave on it using the scenario sidebar (D).

Scenario sidebar

A user can browse the scenarios in the gallery and open a scenario in a detailed view (Fig. 3.4 D). The scenario expands to fill the sidebar with the full user-AI conversation, as well as a brief, AI-generated summary⁶ of the conversation’s contents thus far. As the group reads the scenarios, they start to develop ideas for what to include in the policy.

Formally, a scenario in POLICYPAD comprises three parts: the background (all messages in the conversation up until the most recent turn), the newest user message, and the newest AI message. Given the background and the newest user message, the policy-informed model⁷ generates the newest AI message.

⁶Generated with GPT-4o. More technical details are in Section 3.5.3.

⁷The policy-informed model is an instance of Llama 3.3 70B Instruct. More technical details are in Section 3.5.3.

Interactive in-editor scenario widgets

Users can bring a scenario from their own scenario sidebar into the collaborative editor by referencing its title with the '@' symbol. Once referenced, the scenario appears inline in the editor as an interactive, pill-shaped widget (Fig. 3.5). When a user clicks the widget, the full scenario will be shown in their scenario sidebar. These widgets can be used as illustrative examples of model behavior and build shared context when designing the policy. For example, a user may observe that the model does not provide disclosures of capability limits, or incorrectly assumes a detail not explicit in the conversational context. They can flag a model response in their scenario sidebar, which will make the scenario widget glow orange in the editor, encouraging others to take a look.



Figure 3.5: Scenarios can be brought into the editor inline with the policy as interactive widgets via referencing the scenario’s title with the '@' symbol. Once in the editor, all users can click on it, view it in their scenario sidebar, and flag responses for group discussion.

Drafting policy statements

Once a group reviews the heuristics at the top of the editor to ensure they have a common understanding of desired policy goals, they are ready to start drafting policy statements. These policies address oddities, concerns, and other noteworthy aspects of model behavior they observed and flagged in the scenarios.

As an example, for a policy on providing responsible financial advice, a user may add a policy statement instructing the model to use *cautious*, *neutral*, and *non-prescriptive language* while *always surfacing a brief disclosure of limitations early in the conversation*. A

couple users may collaboratively draft a policy statement for the model to *defer the user to a licensed adviser or a compliant robo-advice product that meets regulatory obligations*. The group can review the policy together and take advantage of the real-time collaborative editing features to further refine each other’s statements.

Testing the policy with scenarios

Users can independently experiment with the behavior of the policy-informed model by *regenerating responses* in the scenario sidebar (Fig. 3.7 1). Independent testing allows each user to focus on the specific concerns that drive their policy contributions, explore challenging boundary cases, and conduct stress-testing without group dynamics influencing their approach. Once they save the policy, they can also browse and compare responses generated by past policy versions.

To propose edits in a non-disruptive way, users can add a **drafting block** (Fig. 3.6) in the editor. Just like how a comment in a code editor is visible to a programmer but does not affect program behavior, content in the drafting block is visible to the group but is ignored by the model. After users review and reach consensus on changes, content in the drafting block can be integrated into the actual policy.

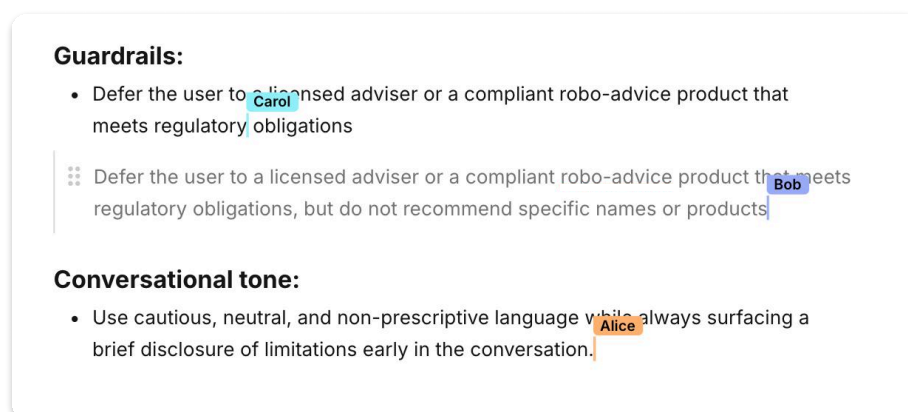


Figure 3.6: A drafting block (directly above “Conversational tone”) can be added into the editor to draft experimental policies without affecting model behavior.

To stress-test a policy, a user can *extend* a scenario in the sidebar by continuing the existing conversation (Fig. 3.7 2). This offers an alternative way to experiment with policy-informed model behavior beyond regenerating a single message in a scenario.

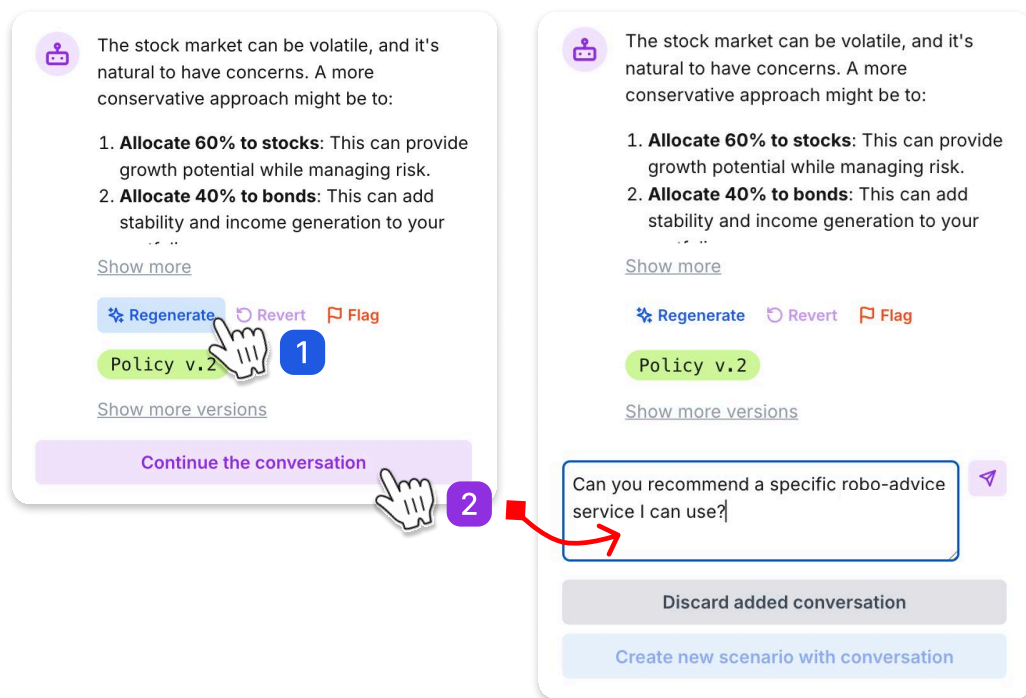


Figure 3.7: POLICYPAD offers two ways to test the behavior of the policy-informed model against a scenario: (1) regenerating the latest AI message, or (2) continuing the conversation.

Group members are not limited to only the scenarios initially provided to them. They may extend an existing scenario and add the extended version to the scenario gallery for others to view and extend further. They may also create a new scenario from scratch if none of the existing scenarios explore a particular behavior they want to test.

After a group has made meaningful edits to the policy, they can **save a new version**. After a user clicks the **[Snapshot policy]** button, POLICYPAD will add the current policy to the version history and regenerate the newest AI message for all scenarios using the policy-informed model (Fig. 3.8 1&3).

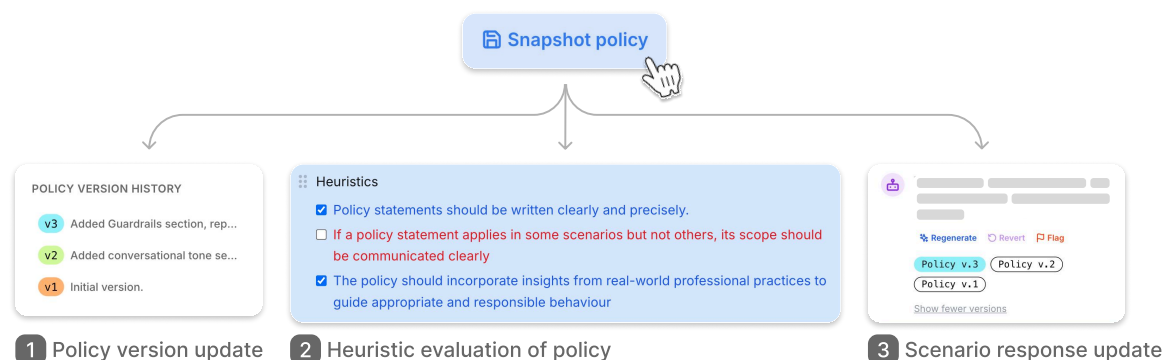


Figure 3.8: Upon saving the policy via the **Snapshot policy** button, POLICYPAD (1) adds the policy to the version history and generates a title summarizing key changes, (2) conducts an automated heuristic evaluation of the policy, and (3) updates the latest responses to all scenarios.

Scenario spotlights

Once a user has referenced a scenario in the editor, they can **spotlight** it to expand the interactive widget into a card UI that makes the full scenario visible to everyone (Fig. 3.9). Once spotlighted, the group can view, discuss, and even collaboratively edit the policy-informed model’s response. Just like other text in the document, the scenario spotlight supports real-time collaboration for editing. After users are satisfied with their edits, any user can save the response. The old response remains easily accessible through a simple toggle.

POLICYPAD then automatically analyzes the group’s edits in the context of the current policy and heuristics. It uses a reasoning LLM (o4-mini) to **suggest a policy statement** designed to steer the model towards producing a response more similar to the edited version (Fig. 3.9 4). If the user accepts the suggestion, it will be integrated into the policy. This alternative way of indirectly editing a policy through editing the model response is inspired by prior work on synthesizing principles from edits [403,524].

Once a group is finished with a scenario spotlight, a user can un-spotlight the scenario for everyone, shrinking the card back into a small, pill-shaped widget.

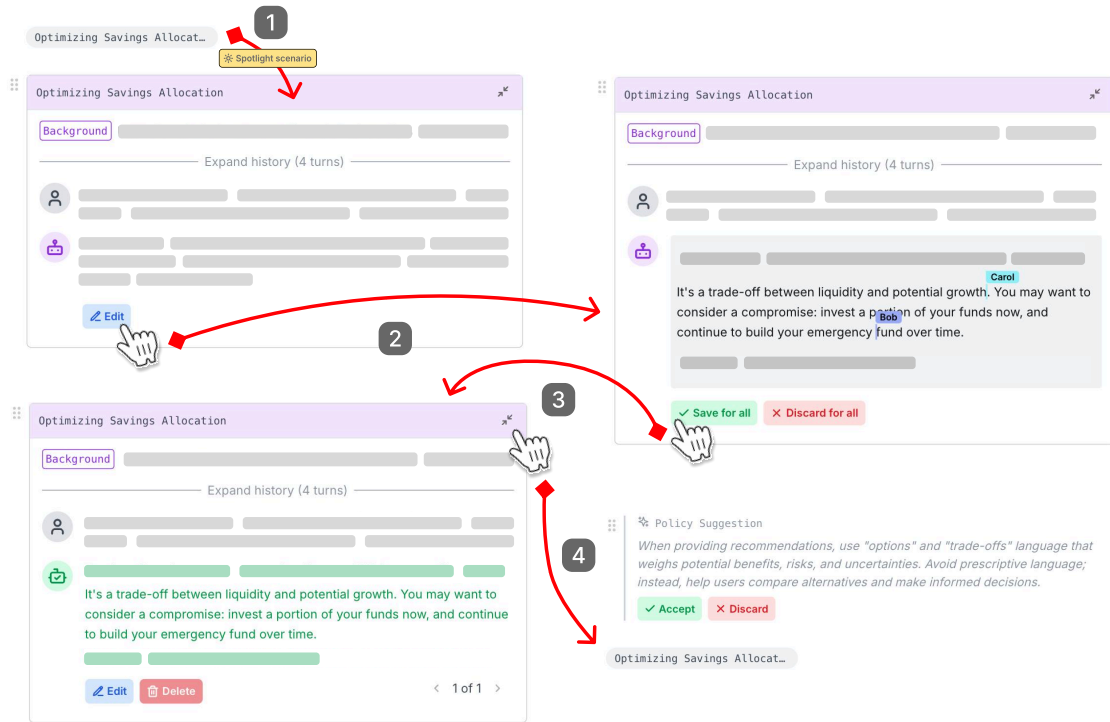


Figure 3.9: Workflow for spotlight scenarios. (1) A user can spotlight an interactive scenario widget to expand it into a card that everyone in the editor can view. (2) The model's response can be edited collaboratively. After saving the edited response (3) and shrinking the spotlight scenario back into an interactive widget (4), POLICYPAD automatically analyzes the edits in the context of the existing policy and heuristics to suggest a new policy statement.

Heuristics editor and evaluator

As a group expands and refines their policy, they may encounter additional considerations they would like to add as policy heuristics. For example, a policy may become increasingly riddled with domain-specific terms and acronyms unfamiliar to a layperson reading the policy, so the group may add a heuristic for clearly defining or explaining these terms. Since the likely audience of this policy is other people, factors like clarity and legibility are important to preserve.

Every time the policy is saved, POLICYPAD runs an automated heuristic evaluation using

o4-mini to highlight any unsatisfied heuristics (Fig. 3.8 2). This automated evaluation is meant to draw attention to the heuristics and encourage discussion around them, rather than conclusively determining their fulfillment. Group members can easily override the automated decision if they agree on a different assessment.

Users continue to engage in this iterative process of policy drafting and experimentation until the policy prototyping session concludes.

3.5.3 *Technical Details*

POLICYPAD is implemented as a web application built with React, TypeScript, and TipTap.⁸ The real-time collaboration engine is supported via TipTap Cloud. POLICYPAD uses serverless functions to call the OpenAI and Together.ai APIs. The policy-informed model is an instance of Llama 3.3 70B Instruct Turbo⁹ hosted on Together.ai. The policy was fed into the model as a system prompt with some additional scaffolding to ensure the model followed it. We called GPT-4o for miscellaneous features that required light LLM processing (e.g., generating titles of new policy versions that capture key changes), and o4-mini for features that benefited from deeper reasoning (i.e. suggesting policy statements after response edits and automated heuristic evaluation). Our prompts are available in our Supplementary Materials.

3.6 *Evaluation Study*

To evaluate POLICYPAD, we ran a series of group-based, within-subjects studies with 22 domain experts from two domains (10 from mental health, 12 from law). Our goal was to determine how the design decisions made for POLICYPAD enhanced the policy prototyping experience. We also evaluate the outputs of POLICYPAD by analyzing the policies created by experts to determine their novelty with respect to established LLM policies like Claude’s

⁸<https://tiptap.dev/>.

⁹The ideal model for policy prototyping generates comparable responses to popular chatbot products (e.g., ChatGPT, Claude) but does not have an existing policy trained into it outside of basic guardrails. Llama 3.3 is an apt fit given its capable performance and bring-your-own safeguards setup (which we did not use). We also used Llama 3.3 instead of Llama 4 due to reports of the latter being narrowly optimized for specific benchmarks [554].

Constitution [19] and OpenAI’s Model Spec [491]. Thus, we asked the following research questions:

RQ1: How did the individual components of policy prototyping supported by POLICYPAD (rapid policy iteration, heuristic evaluation, interaction with scenarios, real-time collaboration) aid expert-driven policy design in practice?

RQ2: To what extent are the insights in experts’ policies created through POLICYPAD novel compared to existing, publicly available LLM policies?

We selected mental health and law as our domains because they are regulated, high-stakes domains for which AI use has been increasing but contested¹⁰ [117, 354, 401, 443]. Crafting responsible LLM policies is therefore critical for ensuring users’ safety and well-being. The two domains are also distinct enough for us to observe how approaches to policy prototyping and the resulting policies can differ across domains. For each domain, we randomly sampled scenarios from datasets compiled by prior work in mental health [354] and law [117]. More details about scenario construction can be found in Section 3.6.1.

3.6.1 Participants and Setup

We recruited 22 domain experts (Table B.3) through our personal connections, university mailing lists, professional Slack channels, and snowball sampling. Among mental health experts ($n = 10$, 7 female, 3 male), the average years of practical experience¹¹ held by each expert was **10.5** (min 3, max 25). Among legal experts ($n = 12$, 5 female, 7 male), the same figure was **5.6** (min 2, max 13). Five of the mental health experts also participated in our earlier observational study.

We organized experts into small groups of 2–4 (median = 3). We observed during our formative study that groups of around three experts struck an ideal balance between creating a collaborative atmosphere and allowing room for meaningful individual contributions.

¹⁰OpenAI CEO Sam Altman observed that the younger generation uses it as a “*therapist, a life coach* [...] asking, ‘What should I do?’” despite the absence of confidentiality protections that govern human attorneys and therapists [518].

¹¹We define “practical experience” as conducting client-facing work at a clinical or legal organization.

Full participant and group details are available in Appendix B.8. Half the groups (4 of 8) contained participants who already knew each other from professional contexts. We did not observe this to impact the quality of discussions or policies prototyped.

We facilitated policy prototyping sessions by providing (with order counterbalanced) POLICYPAD and a baseline system to each group, followed by a brief exit interview. The total length of the study was 90 minutes. Participants were compensated \$150 USD in their choice of cash or a gift card upon completing the study. All studies were recorded and transcribed. This study was classified as exempt by the University of Washington IRB.

Tasks

We assigned each group two policy prototyping tasks, corresponding to distinct sections of a policy. One addressed the **conversational tone**—guidelines for how the model communicates with users. The other addressed **guardrails**—hard-and-fast rules that constrain model behavior for safety and legal compliance. Tasks were counterbalanced by system condition and task order in a 2×2 factorial design.

Baseline system

Our baseline system implemented a simplified policy prototyping workflow that consisted only of iterative policy drafting and experimentation with a policy-informed model identical to the one used in POLICYPAD. It used the same collaborative policy editor as POLICYPAD, but did not include built-in support for interactive scenarios (i.e., the scenario sidebar, scenario widgets, spotlight scenarios) nor heuristic evaluation. It resembled a more polished version of the first prototype used in Section 3.5.1: an editor with a policy-informed model in the sidebar (Fig. B.1). We chose this baseline because it mimicked the system design of many existing “copilot” systems that embed a chatbot in a sidebar alongside a document environment (e.g., Microsoft Copilot or Gemini in Google Docs) as well as policy authoring setups used in prior literature [117].

Starting materials

We prepared scenarios, heuristics, and a small amount of starter text for the policy. For each domain, the first author crafted 10 scenarios that represent a realistic conversation between a human user and an AI chatbot, using the same Llama 3.3 instance as POLICYPAD to generate the responses. To maximize realism without access to product interaction logs, we sourced topics and language from datasets containing realistic questions in our domains of interest: MENTAT from Lamparth et al. [354] for mental health, and `r/legaladvice`-style cases from Cheong et al. [117] for law. We varied the length of scenarios to be 1–5 conversational turns. For multi-turn scenarios, follow-up user messages were written by the first author. For each group of experts, we randomly sampled half the scenarios (5) to assign to the first task, and the rest were assigned to the second.¹² In the system condition, scenarios were loaded directly into the system. In the baseline condition, the first user messages across the 5 scenarios were copied into a Google Doc and shared with participants upon request.¹³

Besides scenarios, we provided three basic heuristics to encourage clear and precise policy writing that draws from real-world professional practices. We also provided an Objectives section in the policy as examples of policy statements, drawn from objectives in OpenAI’s Model Spec [491]. The full starter heuristics and policy are available in Appendix B.6.

3.6.2 Procedure

The studies proceeded as follows:

- *Introduction [5 mins]*: The facilitator introduced the study and agenda, and participants each introduced themselves to each other.
- *Task 1 [30 mins if baseline, 40 mins if system]*: The facilitator oversaw a minimally structured policy prototyping session for either the conversational tone or guardrails.

In the baseline condition, 5 minutes were used for a brief demo. This included some

¹²We note that seeding a policy prototyping session with a different set of scenarios may change its policy outcomes; we discuss how this can be leveraged favorably in Section 3.9.

¹³In the baseline condition, we gave participants the option of starting with or without first browsing these user messages.

time for participants to try out features for themselves. In the system condition, this demo period lasted 15 minutes due to the additional features. The time dedicated to policy prototyping was 25 minutes in both conditions.

- *Task 2 [30 mins if baseline, 40 mins if system]*: The procedure for Task 1 was repeated for a different task in a different system condition.
- *Exit interview [15 mins]*: The facilitator asked each participant to reflect on their experiences across the system and baseline conditions. Participants were also asked to share what they were most excited and concerned about regarding AI use in their domains.
- *Post-study survey*: Group members swapped policies with another group in their domain and rated the policies on 5-point Likert scale questions (see Appendix B.7).

Throughout the study, the facilitator (first author) ensured discussions between participants went smoothly and followed up on specific points when the conversation died down, but otherwise tried to let participants drive the session.

3.6.3 Data Analysis

Thematic analysis (RQ1)

The first author qualitatively coded the study transcripts using reflexive thematic analysis [72, 73]. An initial deductive pass isolated specific parts of the transcript that were highly relevant to each research question, followed by one or more inductive passes to surface themes organically. This analysis was augmented by short memos the facilitator wrote upon concluding each study, summarizing key events and noteworthy insights from each session.

Policy novelty analysis (RQ2)

To analyze the novelty of policies—whether they contribute new perspectives, ideas, considerations, dependencies, or approaches to existing policies—we gathered experts’ policy

statements from all sessions and evaluated each against publicly available policies for guiding responsible model behavior (the “existing set”). We combined all policies from OpenAI’s Model Spec [491], Claude’s Constitution [19], and principles derived from workshops with legal experts in prior work [117], to represent the set of existing policies.¹⁴

Rather than rely on direct human coding of novelty between policies, we opted for a joint human-AI approach where we made use of LLMs to first identify portions that were *likely to be novel* before having human annotators review and make the final novelty determinations. Specifically, we followed this procedure:

1. For each expert-written policy statement, we used 3 prompts with varying definitions of novelty¹⁵ to make binary novelty decisions against the existing policies. Our prompts also required the LLM to generate a justification for its decision.
2. Any policies that were not unanimously determined to be novel in all 3 prompt evaluations were considered not novel. For the remaining policies, we further prompted the model to retrieve relevant quotes from the existing policies to be used as context for human evaluation.¹⁶ Retrieved quotes were examined by the first author alongside the full existing set to check for hallucinations and any missed quotes.
3. Finally, two human annotators (members of our research team) reviewed the list of policies and quotes to make a final novelty determination. Annotations were first done independently, with an initial Cohen’s Kappa of 0.41. Disagreements primarily arose from differences in the interpretation of novelty; they were resolved via a round of discussion. If annotators failed to reach a consensus (which was the case for 2 policy statements), the policy statement was considered not novel by default.

While the reliability of LLMs for making content judgments has been called into question by recent work [106, 319, 341, 636, 643], we structured our evaluation process to minimize

¹⁴Excluding policies specifically targeted at moderating hate speech and disturbing content (e.g., [353]), as they are orthogonal to the policies we focus on in the prototyping sessions.

¹⁵Executed on GPT-4.1. Prompts provided in Supplementary Materials.

¹⁶Also through GPT-4.1.

the potential impacts of these factors through deliberate prompt design. Specifically, we attempted to control for model sensitivity to prompts by using multiple variations of prompts for initial evaluation. Additionally, we incorporated existing policies from prior work [19,117,491] into the prompt and requested justifications and quote extraction to identify possible model hallucinations. Finally, we ensured the final novelty determination was done by human annotators. Overall, we believe that this process should yield a *conservative* determination of novelty, while also ameliorating challenges around human attention during review and comparison of exceptionally long texts. Our prompts used in this evaluation are available in our Supplementary Materials.

3.7 Findings

3.7.1 Design Decisions in POLICYPAD Fostered Collaboration During Policy Prototyping (RQ1)

Heuristics built common ground and inspired richer policies (system only)

Participants generally agreed that having heuristics as part of the policy prototyping process helped develop common ground for the group. P3 found that heuristics helped the group align on “*the spirit of what we were doing*” and ensure “*we’re on the same page about the purpose of the policy.*” P19 had a similar experience: “*[the heuristics] set the underlying tone for how the policy is supposed to function.*” P5 thought the heuristics gave “*an idea of what sorts of [policy statements] would work best,*” while P4 agreed, finding that heuristics offered “*more specific guidance*” for drafting policies around edge-cases in model behavior. P7 appreciated heuristics as “*a constant reminder of the guidelines,*” but recognized that because domain experts are already well-acquainted with many of these guidelines, heuristics might be even more useful for developers who are refining the policy and integrating them into models.

Besides serving as guidelines, heuristics also served as entry points for deeper discussion on key policy topics. The starter heuristic on incorporating real-world professional practices into the policy initiated discussions in MH01, MH02, and MH04 about *motivational interviewing* (MI), a foundational technique for therapists. Experts then incorporated various

aspects of MI into the policy, such as encouraging “*summaries of conversation when appropriate*” (MH01), and “*Repeating or paraphrasing what [the user] is saying*” (MH02). MH02 and MH03 brought up *limits of confidentiality*—when a mental health expert is legally or ethically required to break confidentiality to share client information, even though expert-client conversations are otherwise private. MH02 brainstormed situations in which experts needed to break confidentiality (e.g., “*immediate risk of harm to oneself or others; suicidal thoughts, urges, or behaviors; presence or risk of non-suicidal self injury*”) and added a policy to “*avoid using MI*” in those situations. MH03 agreed that models, just like when experts work with clients, should provide a disclaimer early on in the conversation of conditions under which confidentiality will be broken and remind the user that “*[the conversation] is not a confidential setting*” when those conditions are triggered.

Interestingly, a couple groups of legal experts used the heuristic to debate whether real-world practices for lawyers and other legal professionals should even apply to AI, since they clearly established that AI does not have the same legal status as human lawyers. P17 in L02 shared that while “*we lawyers do have rules of professional responsibility that we need to adhere to, they don’t apply to non-lawyers*” and suggested removing that heuristic. P20 in L03 echoed that sentiment: “*Those ethical rules of lawyers do not apply to AI systems.*” Their group member, P19, agreed, and noted that “*if ethical rules of lawyers did apply, then the AI model cannot even begin to suggest answers.*” In general, this is an important point of distinction between the mental health and legal policies; we unpack this in more detail in the Discussion (Section 3.8).

We observed that groups did not initially modify the starter heuristics provided to them, but some added more heuristics as they worked on their policy. For example, group MH02 realized their policy contained some mental health-specific concepts that needed to be explained to the facilitator, and added a couple heuristics to “*Give illustrative examples for concepts and terms*” and “*Give definitions for jargon and technical terms.*” Similarly, L03 added a heuristic to “*Clearly explain or define legal terminology.*” As their policy got longer, L02 added a heuristic to ensure “*No policy statements should conflict with each other.*”

Spotlight scenarios improved collaborative dynamics and provided valuable writing support (system only)

Participants found the ability to bring scenarios into the editor, spotlight them, and collaboratively edit their responses to be valuable features in POLICYPAD. In general, we observed a pattern where participants referenced scenarios in their discussions and then referenced them in the editor for others to view. P22 said the ability to “*input the scenarios into the editor helped with brainstorming and being able to point to specific parts of a response we either found helpful or that we thought needed to be changed.*” P5 agreed and thought that the utility of spotlight scenarios could scale with the number of collaborators: “*[the spotlight] would be really nice for larger groups of people contributing, being able to look at [the scenario] together.*” P1 thought spotlight scenarios can be useful for facilitating asynchronous collaboration as well: “*I see [P3] has already edited this side of things. I can hop right back in and draft out a version of the [policy] and stress test it.*” After using the interactive scenarios, P11 thought that “*for a collaborative effort, [POLICYPAD]’s really nice. [The baseline] felt more like a personal tool.*”

Indeed, in the baseline condition, whether it came before or after the system condition, experts were finding makeshift ways to accomplish what scenario spotlighting is designed for. For instance, P4 asked to share their screen so everyone could view the policy-informed model response they generated. Similarly, when P8 was pointing to a specific aspect of a generated response, their groupmate P7 asked: “*Is that something you can share so we can all see it, so we just work off of that one?*” In both cases, experts improvised a solution by using a **drafting block** to share content from scenarios in the editor without impacting the policy.

Participants also expressed appreciation of the ability to edit the response and receive a system-generated policy suggestion. 100% of the policy suggestions were accepted in our studies. P17 shared that the suggestions were helpful in articulating their thoughts: “*sometimes it’s difficult to put your thoughts into words, and the [suggestions] are helping you with that.*” P10 agreed, saying that “*the ability to pull that response in, edit it, and have a generated guardrail could be a huge time saver.*” They saw it as a way of removing the

need for low-level wordsmithing: *“We don’t need to edit that response perfectly, but if we can make it clear what our priorities are, and then see if the AI gets our nuance, that’s pretty incredible.”* We also observed that in MH04, P9 and P10’s policy drafting began slowly, but accelerated considerably when they received policy suggestions that inspired more ideas. P7 shared why they accepted policy suggestions even when they seemed imperfect: *“I liked the policy [suggestions], even if they weren’t necessarily dead on. It gave us ideas for other [policy statements].”*

Experimentation with a policy-informed model directly informed policy edits (system & baseline)

In both the system and baseline conditions, we observed how quick and iterative experimentation with the policy-informed model¹⁷ benefited the policy prototyping process. Experts easily surfaced specific model behavior that could be addressed with the policy, such as when the model was overstepping its role, such as making a judgment about *“whether a risk is worth or not worth taking”* (P13). Experts could then draft the policies and immediately observe the impact their edits had on the response, either by clicking the [Regenerate] button for quick testing or taking a snapshot of the policy and updating all responses to all scenarios at once. As experts critically evaluated the responses for common behavior they targeted in the policy—such as judgmental language (L01, L02, L04, MH01, MH02, MH03), eliciting necessary information from users in order to provide a responsible answer (L01, L02, L03, MH01, MH02, MH04), and the inclusion of disclaimers (all groups)—they could qualitatively observe clear improvements. For example, at the end of their session, P19 confirmed that *“I see everything we’ve discussed being implemented, and [the model] still manages to give a fair amount of information, so that’s good.”*

Real-time collaboration amplified other benefits (system & baseline)

Participants actively engaged with each other during the sessions—seeking and providing peer feedback, discussing nuances and complexities of model behavior, sharing insights from

¹⁷Recall from Section 3.6.1 that the baseline uses the same model as POLICYPAD.

their own professional experiences, and more. This engagement benefited all components of the policy prototyping workflow.

They provided input for and helped edit responses when a group member put a scenario on spotlight. They shared explorations of edge cases in model behavior with the group to patch gaps in the policy and identified high-risk scenarios to focus their discussions. Overall, real-time collaboration amplified participants’ abilities to draw upon their expertise, challenge assumptions, and iteratively refine policies.

We observe that participants tended to agree with others in their group and rarely challenged or pushed back directly on others’ input. This may be due to a desire to appear diplomatic and accommodating, especially when working with new collaborators.

3.7.2 Experts Prototyped More Novel Policies in POLICYPAD Than the Baseline (RQ2)

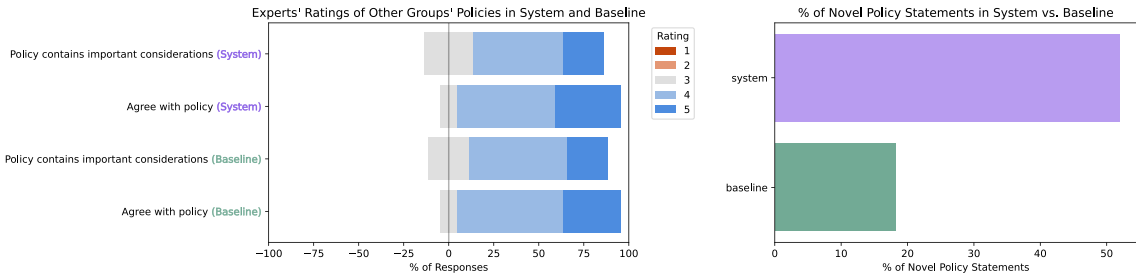


Figure 3.10: Comparison of Likert scale responses to policies prototyped in the system vs. baseline (**left**) and the novelty of policy statements prototyped in the system vs. baseline (**right**).

Quantitative results

Our novelty analysis (Section 3.6.3) revealed that experts prototyped more novel policies using POLICYPAD compared to the baseline (Fig. 3.10 right). **51.9%** of the policy statements drafted in POLICYPAD were considered novel, compared to **18.2%** from our baseline. Looking at raw numbers, the number of novel policy statements from POLICYPAD was **4 times** that of the baseline (40 vs. 10).

Outside of novelty, we found the policies from the two systems to be comparable (Fig. 3.10 left). After the study, experts rated policies from another group within their domain along two dimensions: the extent to which 1) the policy contained *important considerations* of AI behavior within their domain, and 2) they *agree* with the policy. Wilcoxon signed-rank tests on Likert data showed no significant difference ($W = 6.0; p = 0.65; M_{system} = M_{baseline} = 4$ for important considerations, $W = 20.0; p = 0.74; M_{system} = M_{baseline} = 4$ for agreement). This suggests that experts generally viewed policies from other groups favorably, and that novelty was the main differentiator between policies across the two systems. We suspect the high positive bias may stem from collective alignment on priorities when serving clients, which in turn may be due to some standardization in professional training (e.g., motivational interviewing in mental health).

Qualitative results

We conducted a qualitative analysis of the novel policies to understand what exactly was novel about them. We identified three main sources of novelty.

First, experts’ policies offered more insight into **specific circumstances under which the model should defer the user to a human expert**. MH01 noted that while the model can provide empathy and reassurance, as soon as indications of “*behavioral interventions (such as behavioral activation [for depression], exposure and response prevention [for OCD], or prolonged exposure [for PTSD])*” arise, the model should defer to a human therapist. L03 shared that the user’s desire to share confidential information with the model is a good indicator of whether the model should defer to a legal expert: “*If the user indicates that they want or need to provide confidential information, there may be privileged information involved. If the conversation contains privileged information, always defer the conversation to a legal expert.*” L04 summarized their perspectives on this issue as: “[*The model should*] answer ‘*what can I do*’ questions and defer ‘*what should I do*’ questions to a lawyer.” Generally, current models’ lack of awareness of when to defer to human experts is a significant safety concern [252], and experts’ policies have potential to improve this awareness.

Second, experts’ policies provided **specific procedural guidelines** that existing poli-

cies lack or do not specify in much detail. MH01 and MH02 both provided guidelines for motivational interviewing in their policies, but noted that the technique should not be used in high-risk situations. In such situations, MH04 emphasized the model should be much more succinct and direct when supplying crisis response information: *“When a user indicates a high level of distress, or [when] crisis services or hotlines are required, provide them succinctly and without much additional text.”* MH03, L02, and L03 all instructed the model to disclose limits to or the lack of confidentiality early in the conversation. MH03 specifically stated that after this disclosure, the model should then *“Ask users if they have questions about confidentiality limits,”* reflecting a procedure in their own mental health practice.

Finally, experts’ policies recommended that the model be **more proactive about seeking key information at the start of the conversation**. Experts also considered it irresponsible for the model to assist users when lacking key information, such as legal jurisdictions. L03 wrote that the model should *“require the user to indicate their jurisdiction before providing full responses.”* Similarly, L02 recommended that the model avoid providing assistance if it cannot elicit *“essential case details (such as date of offense, location, and current legal status) necessary to tailor legal guidance.”* MH01 and MH04 both wanted the model to conduct a more thorough risk assessment prior to engaging deeply with the user. MH01 recommended that the risk assessment include *“asking how problems or challenges have been addressed (or not addressed) before, how long the problem has persisted, and how distressing/problematic the user finds the current situation.”* These policies can help augment existing technical efforts in improving LLMs’ abilities to elicit user information to improve their quality of assistance [14,380].

3.8 Discussion

3.8.1 Practical Deployment Considerations

Results from our policy prototyping sessions showed that experts can contribute novel policies to shape model behavior in domain-specific, high-stakes interactions with users. As these interactions become more commonplace, it is crucial for users’ safety that model behavior is vetted, scrutinized, and refined by expert input. Policy prototyping is one promising avenue

for this, but numerous considerations for widespread adoption remain. These include resource constraints, time commitment, learning effort, and impacts on experts’ own careers. We now discuss these considerations and some possible ways forward.

Time and resource intensity

First, policy prototyping can be time- and resource-intensive. Our sessions were 90 minutes in length and could have easily been longer if not for scheduling constraints. Our sessions were one-off events, but LLM policy design can and should benefit from *sustained* expert engagement [493, 660]. Challenges associated with time and resource intensity may be alleviated if policy prototyping was more seamlessly integrated into experts’ professional practices and did not cold-start like it did in our work. After all, academic or industry labs are not the only ones motivated to design better LLM policies—expert communities *themselves* are too. As AI diffuses into their domains, they are incentivized to take bottom-up approaches¹⁸ to shape model behavior, due to its ability to impact the behavior and actions of clients [117, 253]. Indeed, the American Psychological Association (APA) has started to track the use and integration of AI into mental health practice as a key industry trend [1]. APA and similar organizations may thus be interested in hosting policy prototyping sessions at gatherings or conferences, potentially in partnership with AI developers, as a way of actively engaging the community to shape AI diffusion in their domain. Additionally, local chapters may have the capacity to host sessions more regularly as part of ongoing discussions.

Expert displacement

Relatedly, it is reasonable to think that experts’ efforts in policy prototyping could result in AI taking over large portions or even all of their work. A therapist, for example, might design a policy that faithfully reproduces how they interact with clients, only to find that the AI could then offer a cheaper substitute for their services. This possibility highlights an important benefit of policy prototyping: *it equips experts with a voice to shape how AI*

¹⁸Assuming that model developers are open-minded about integrating at least some of experts’ suggestions.

can impact their work. Instead of reproducing therapeutic behavior, mental health experts can encode guardrails that push the model away from emulating them directly. Indeed, several groups adopted this approach in our study (see Appendix B.9). As long as there are incentives for model developers to adopt policies created from third-party input, which prior work has shown to be the case [268, 485], policy prototyping offers experts more agency to guide how AI diffuses throughout their domains.

Organizations for facilitation

Finally, who can facilitate policy prototyping sessions? In this work, the sessions are facilitated by an academic lab interested in studying expert-informed LLM policy design. Being affiliated with a university allowed us to take advantage of being in close proximity to—or already connected with—experts (e.g., in academic departments, the medical school, the law school, etc.). However, in practice, we see a wide range of organizations with diverse resource profiles that can collaborate with each other for even more effective facilitation. For example, non-profits interested in AI safety, such as METR¹⁹ and the AI & Democracy Foundation,²⁰ alongside organizations advocating for responsible AI more generally (e.g., Partnership on AI,²¹ Ada Lovelace Institute²²), can set agendas for policy prototyping that target emerging harms and underrepresented stakeholder concerns. National AI safety centers (e.g., UK AISI,²³ US CAISI²⁴) can guide policy design efforts to prioritize legitimacy and trust in the public interest. Safety and alignment teams within frontier model development companies (e.g., Anthropic, OpenAI) can leverage resources for expert recruitment and contribute technical tooling. Ultimately, effective facilitation will likely require combining these strengths across institutions rather than relying on any single actor.

¹⁹<https://metr.org/>

²⁰<https://aidemocracyfoundation.org/>

²¹<https://partnershiponai.org/>

²²<https://www.adalovelaceinstitute.org/>

²³<https://www.aisi.gov.uk/>

²⁴<https://www.nist.gov/caisi>

3.8.2 *Scaling Up Policy Prototyping for Participatory Model Behavior Design*

Our work focused on engaging small groups of domain experts in deliberating on and actively creating policies for model behavior. We see potential in broadening our approach beyond collaborative design with experts. This is valuable when democratic participation is prioritized over narrowly defined expertise, or for topics where the notion of “expertise” is blurry (e.g., personal information management). To expand beyond experts and reach towards more *participatory* visions of AI development with diverse stakeholder groups [48,126,141,303,634], efforts to scale up policy prototyping—including tool ecosystems that do not only consist of POLICYPAD—will be required.

A challenge that will inevitably arise with increasing scale is resolving disagreement. We have already started to see this challenge emerging within our small-scale groups—expert groups within and across domains did not always agree on how to design the policy. Within *mental health groups*, there was some disagreement over 1) whether the model should act like a therapist, and 2) the appropriate conversational tone before a proper assessment of the user is made. Within *legal groups*, experts disagreed over whether the model should suggest action items for the user—a behavior that was found to be unanimously criticized by legal experts in prior work [117]. *Across the domains*, disagreements arose over whether the model should, under any circumstances, attempt to mimic a human professional. More details about these disagreements can be found in Appendix B.9.

Prior work has attempted to resolve disagreements on AI behavior at scale through multiple rounds of voting and asynchronous deliberation [268]. However, asynchronous collaboration methods rarely uncover the same richness and nuances of disagreements that we observed in our synchronous, deliberative policy prototyping sessions. To scale policy prototyping to include more stakeholders while preserving the quality of discussions, we may draw inspiration from multi-stage or tiered citizens’ assemblies (also known as “mini-publics”) [126,423,512]. These systems aggregate deliberations from parallel, local assemblies into regional or (inter)national assemblies for producing recommendations. For policy prototyping, parallel deliberations may be held on a regular basis with groups of individuals whose personal and professional lives are impacted by model behavior so they have more

agency to shape it (e.g., parents who are concerned about their children interacting with AI companions, workers who are expected to use AI tools to boost their productivity). The resulting policies can then be aggregated with policies prototyped by domain experts in parallel. Aggregations can also happen across geographic regions to incorporate pluralistic cultural and social perspectives.

New suites of tooling and infrastructure will likely be needed to facilitate policy prototyping at scale. At the deliberation level, tools like POLICYPAD can help. At the aggregation level, civic technologies like Pol.is [606] are more suitable. We encourage future work to explore different combinations of collaborative affordances and interaction paradigms to build new tool suites for scalable policy prototyping.

3.8.3 *Situating Policy Prototyping Within AI Alignment*

We proposed policy prototyping as a practice through which small groups of policy designers can collaboratively design LLM policies. Where can this practice fit within the broader AI alignment pipeline?

A prerequisite for policy prototyping is an instruction-tuned model with behaviors representative of what users will experience in the wild. This is most commonly a frontier model—a model with frontier performance on popular benchmarks—as they are commonly integrated into user-facing applications. Thus, we envision policy prototyping taking place *after* fundamental alignment and safety efforts (e.g., instruction tuning, RLHF, implementing basic safety guardrails and classifiers, automated red-teaming). However, policy prototyping should come *before* later-stage or more sophisticated alignment efforts that benefit from or even require a policy (e.g., deliberative alignment [235], manual red-teaming [4]). If a developer has an existing policy, policy prototyping can reveal nuances, inconsistencies, and areas needing refinement before the developer commits significant resources to align the model with it. If the developer does not yet have a policy, policy prototyping can help start one.

We also note that LLM policies are continuously evolving artifacts, rather than static ones [491]. Model developers may thus find it helpful to *co-evolve* the policy and alignment

strategies. This can involve policy prototyping sessions with experts on a regular basis to seek input on top-of-mind concerns based on insights from usage telemetry.

3.9 Limitations and Future Work

All our participants except for two legal experts were based in the US. The perspectives integrated into our policies are thus heavily influenced by the American mental health and legal systems, and may not generalize to other countries. Before integrating these policies into AI systems that serve a global userbase, future work should augment and contrast our policies with perspectives of non-US and non-Western experts.

The selection of scenarios for policy prototyping can alter policy outcomes by steering discussions towards topics depicted in the scenarios. We selected scenarios based on realistic tasks experts in mental health and law might tackle in their day-to-day work, but our scenarios may not achieve adequate coverage over the diversity of scenarios experts actually encounter. We also manually composed responses to the LLM in multi-turn scenarios, which may not reflect real-world responses from users asking about mental health or legal questions. Further, we only started with 5 scenarios per session to fit the study within the 90-minute time limit. Future work can explore the optimal number of scenarios for a certain group size, as well as dedicating time for group members to author scenarios prior to prototyping the policy to further improve realism.

Researchers running future policy prototyping sessions may be interested in potential modifications of our setup. We recommend exploring three modifications. First, the facilitator for a workshop holds a major role and can influence the results with their facilitation style. The first author facilitated all sessions in our study for consistency, but future work can experiment with different facilitation styles to determine which are more effective. Second, the starting scenarios can focus on a particular theme or issue within a domain for more targeted policy design. For example, due to recent high-profile cases of AI-driven psychosis [247, 253, 254], scenarios can draw from real transcripts of psychosis-inducing conversations [253] rather than our random sampling approach. Third, the sessions can be scaffolded with taxonomies of concepts within a specific domain. Prior work relied on concepts as a central ingredient in policy design [353]. While concepts were not fundamental to

our work, including them may benefit future sessions.

Finally, experts agreed the model should elicit key information from users before providing assistance (Section 3.7.2). Effective elicitation requires the model to *reason about missing information*. While it is promising that LLMs’ reasoning capabilities have improved significantly in recent months, improvements have primarily focused on verifiable domains like math and coding [453], and it is unclear whether these improvements translate to more effective information elicitation. Future work can empirically investigate this and develop techniques for models to reason about missing information in contextual, human-centered ways.

3.10 Conclusion

In this work, we asked: *How can domain experts be meaningfully involved in designing LLM policies as a means of actively shaping responsible model behavior?* In response, we introduced POLICYPAD, an interactive system for small groups to engage in LLM policy prototyping—a practice that draws upon UX prototyping methods to enable collaborative drafting, testing, and rapid iteration of LLM policies in real time. We conceptualized LLM policy prototyping and motivated the design of POLICYPAD through a 15-week observational study with 9 mental health experts. We then evaluated POLICYPAD through 8 policy prototyping sessions with 22 experts in mental health and law. We found that POLICYPAD fostered a collaborative and productive dynamic for policy prototyping and led to the creation of more novel policies compared to a baseline. Areas of novelty covered important considerations for model behavior, such as when to defer to human experts, specific procedures for emergency situations, and eliciting missing information needed to responsibly provide assistance. We hope future work will extend our contributions and continue co-designing policies with experts and laypeople alike to improve the safety and responsibility of advanced AI.

Chapter 4

CANVIL: DESIGNERLY ADAPTATION FOR LLM-POWERED USER EXPERIENCES

POLICYPAD demonstrated that model behavior can be collaboratively prototyped in a document-based environment by groups of users. This chapter builds on that work by exploring how designers can integrate similar prototyping practices into their design workflows. Model behaviors governed by policies from POLICYPAD are by default decoupled from any specific application. Designers, on the other hand, are often working in the context of specific applications with existing design requirements. This chapter introduces CANVIL, a Figma widget that operationalizes designerly adaptation, a process through which designers engage with LLMs as a design material and co-evolve design requirements and model behavioral specifications.

4.1 Introduction

Since generative AI came under the public eye in late 2022, a shift has been underway for integrating generative AI capabilities into everyday user-facing technologies. Large pre-trained models, most notably large language models (LLMs), have versatile natural language capabilities that unlock novel interactive techniques and interfaces for more intuitive and customizable user experiences across a wide spectrum of applications [16, 511, 629, 688]. However, these promises also come with numerous concerns. Integrating LLMs into a domain without careful consideration of the user contexts surrounding model use and implementation of appropriate guardrails may result in user experiences that perpetuate societal biases [570], threaten users' sense of well-being [556, 584], or otherwise do harm [99, 151, 580, 591, 679].

As technology development practitioners, designers¹ are uniquely positioned to mitigate

¹In this chapter, we consider designers to be anyone who has an active, hands-on role in designing the user experience of a product and/or feature. This includes job titles such as UX designer/researcher, content designer, and product designer. This contrasts with the broader definition of designer used in the rest of this dissertation, as noted in Chapter 1.

these concerns [183, 385, 625, 699, 711] and operationalize growing calls for human-centered AI [355, 601]. Designers’ work often involves aligning technological capabilities with user needs, such that the technology addresses (or makes progress towards addressing) pain points identified in user research [183, 623]. Designers are trained in human-centered design methods that allow them to understand users and usage contexts, prototype potential solutions with relevant technology, and iterate on those solutions based on their understanding of users or user feedback [42]. Yet, prior work has shown that designers face diverse procedural challenges (e.g., difficulties collaborating with the engineering teams training the models [623, 626]) and instrumental challenges (e.g., lacking means to effectively work with the models [183, 625, 699]) when working with AI [626, 701]. In efforts to address these challenges, researchers have situated AI as a design material to highlight its material properties—including the technology’s capabilities, limitations, and adaptability [377]—for designers to better understand and apply AI in the context of their design problems [154, 181, 699]. A good *designerly understanding* of AI can help designers ideate on new AI-powered design ideas, mitigate AI’s varying impact for different user scenarios, collaborate with design and non-design practitioners, and reinforce user-centered perspectives amongst the team and throughout the product development cycle [385, 664].

The advent of powerful pre-trained LLMs introduces new opportunities for engaging with AI as a design material. First, adaptability emerges as a key materialistic property of LLMs—these models are responsive to adaptation via fine-tuning [41, 152, 267, 499] and prompt-based methods [476, 477, 687]. In fact, due to resource-intensive training of LLMs, it has become a standard practice for individual developers and development teams to adapt “base” LLMs from a small handful of providers (e.g., Anthropic, Google, OpenAI) instead of training custom in-house models [511]. Moreover, natural language interaction and adaptation democratize AI experimentation for practitioners—including designers—traditionally excluded from AI conversations due to limitations in technical expertise [385, 521]. Despite this, there has been limited exploration of designers’ interaction with—let alone adaptation of—LLMs in practice as they contribute to LLM-powered features and products.

In this chapter, we first conduct a formative interview study with 12 designers experienced in designing features and products that use LLMs to better understand their workflows

and desiderata when crafting LLM-powered UX. From our interviews, we identify a need for a translational process that allows design requirements (e.g., findings from user research, product goals) to directly shape model behavior, and vice versa. Drawing from Cross’ notion of *design codes*—systematic ways of representing, transmitting, and translating design knowledge [134]—we propose *designerly adaptation for LLMs* (henceforth “designerly adaptation”) as a new design code to facilitate this two-way translation. We operationalize designerly adaptation by building CANVIL, a Figma² widget that supports structured authoring of model behavioral specifications based on design requirements to adapt LLMs, testing of adapted models on diverse user inputs, and integration of model outputs into interface designs. We then use CANVIL as a technology probe to study designerly adaptation with 6 groups with a total of 17 designers. We demonstrate the effectiveness of our probe by collecting concrete, empirical insights into how designers engage in designerly adaptation through CANVIL. Specifically, we find that designers iteratively translated diverse design requirements into LLM behavioral specifications. This allowed them to simultaneously adapt their designs to enhance user interaction with LLMs, while setting or refining behavioral constraints for the model using their designs. Furthermore, designers recognize promises in collaboratively engaging with designerly adaptation to share resources and knowledge with design and non-design stakeholders. They are optimistic about integrating designerly adaptation into their own workflows and note additional procedural and instrumental questions to address in practice. Concretely, our work makes the following contributions:

- Designerly adaptation—a process bridging material exploration of LLMs with design processes—motivated by insights from a formative study with designers who craft LLM-powered UX.
- CANVIL, a technology probe to operationalize designerly adaptation in the form of a Figma widget and allow us to study it in practice.
- Empirical learnings uncovered by CANVIL about how designers engage in designerly adaptation in a task-based design study.

²Figma is a popular collaborative design tool: <https://www.figma.com/>.

- A discussion of our learnings’ implications on design practice and beyond, including a proposed workflow for designerly adaptation to orient future work and a reflection of LLMs’ sociomaterial implications on collaborative practices within product teams.

4.2 *Related Work*

Chapter 2 surveyed AI as a design material and object of adaptation (Section 2.2.2), along with the methods and means for shaping AI behavior (Section 2.2.4). Here, I position our work against research closest to designerly adaptation.

4.2.1 *Designerly Adaptation of LLMs*

Because of LLMs’ powerful capabilities, there is significant interest in exploring their integration into user-facing applications [355,511]; as such, designers should be prepared to work with them as a design material [343]. While past work advocated for a better “designerly understanding” upon studying how to support designers in working with models [385], our work specifically explores involving designers in the practice of adapting LLMs. Adaptation and understanding can be related and synergistic, but our focus is on adaptation—a common practice in product teams [511]—and how it can support designers to craft LLM-powered UX.

In our work, we examine and support *designerly adaptation* of LLMs—a means of empowering designers to explore LLMs as a design material through model adaptation. Unlike many technical adaptation methods, designerly adaptation contributes to human-centered AI by drawing from designers’ workflows and expertise. This focuses efforts to ensure that AI-powered applications satisfy user needs and effectively co-adapt with us as they become more pervasive. While we draw from recent developments in and recommended best practices for accessible model adaptation via natural language, we acknowledge that these practices may shift over time as we co-adapt to this new technology. Using currently available techniques is just one possible instantiation of designerly adaptation.

Despite advancements in tooling, support for designers to work with LLMs as an adaptable design material remains limited. Domain-agnostic tools for tinkering with models may not be well-integrated into design workflows, a primary consideration for designers when

deciding whether to adopt those tools [182]. Tools that offer integration with design environments (e.g., PromptInfuser [521,522]) support prototyping with LLMs, but not necessarily deeper adaptation of model behavior. Our work situates interactive adaptation within familiar design environments and processes.

4.3 *Formative Study*

Working with AI, especially LLMs, is an emerging practice in the field of UX [181,626,701,712]. We conducted a formative interview study to understand designers’ experiences working on LLM-powered products and features amidst industry-wide shifts towards LLMs in 2023 [511] as an initial step in our research.

4.3.1 *Method and Participants*

We conducted 30-minute 1:1 virtual interviews with 12 designers at a large international technology company where LLM-powered features and products are actively explored. Our interviews were not initially focused on model adaptation, let alone designing a probe for adaptation. Instead, our goal was to more deeply understand designers’ workflows and pain points they face when crafting LLM-powered UX. Thus, we opted for interviews as our method of choice. All interviews were conducted in June and July 2023. At the time of the study, all participants were working on products or services that leveraged LLMs in some capacity; LLM application areas spanned conversational search, question-answering (QA) on domain-specific data, recommendation, text editing and generation, and creativity support tools. Participant details can be found in Table A.1 of Appendix A.1.

Our interviews were semi-structured and revolved around the following topics:

- **Awareness:** to what extent are designers aware of LLMs’ capabilities, limitations, and specifications in the context of their product(s)?
- **Involvement:** to what extent are designers involved in discussions or activities that shape where and how an LLM is used in their product(s)?

- **Desiderata:** what do designers desire when crafting LLM-powered UX, with regards to both processes and tools?

Each participant received a \$25 USD gift card for their participation. All interviews were recorded and transcribed. Our study was reviewed and approved by the company’s internal IRB.

The first author performed an inductive qualitative analysis of the interview data using the qualitative coding platform Marvin³. This process started with an open coding round in which initial codes were generated, followed by two subsequent rounds of axial coding in which codes were synthesized and merged into (yet-to-be-named) groups, in line with Braun and Clarke’s reflexive thematic analysis [73]. These groups were reviewed and named through an iterative process that involved the disassembling and merging of groups; the named groups became initial themes. The codes and themes were discussed with research team members at weekly meetings. Other research team members also offered supporting and contrasting perspectives (e.g., (dis)agreement on main themes, interpretation of new themes) by writing their own analytical memos, which were also discussed as a team. Any disagreements were resolved through team meetings and asynchronous discussion via comments on the analytical memos.

4.3.2 Findings

Adaptation was seen as a central materialistic property of LLMs

Participants recognized that products delivering compelling, robust UX were not powered by out-of-the-box “base” LLMs, but required adaptation based on user needs and contexts. P5 gave an example where an out-of-the-box model inappropriately prompted the user for a riddle in a workplace setting: “[*The experience*] is not quite right, cause it’s in [*workplace software*] and it’s like, tell me a riddle. I’m at work!” P4 and P7 both stated a design goal for their products (in conversational search and domain-specific question-answering, respectively) is to accommodate non-technical users, which can be achieved if the model can

³<https://heymarvin.com/>

“tailor the language depending on technical ability, and gradually introduce [users] to more complicated concepts and terminology” [P7]. P2 shared that their team settled on using three adapted versions of the same model to customize UX within their product: “There are actually three incarnations of the same model, each with slightly different parameters [...] based on the prompt that users give, we select which of those 3 incarnations we wanna use.”

In their design work, participants appreciated the ability to steer an LLM with natural language to envision more flexible and adaptable AI-powered UX. Some were familiar with how lightweight adaptation techniques, such as writing system prompts, can be used to specify model behaviors based on design specifications. P12 discussed an example of how this may work in an entertainment system setting:

“You might want to generate enthusiasm more, right? So the LLM that goes over there might have a UX layer that feels different. The system prompt can say, remember, you’re a machine that wants to get users enthused.” [P12]

P12’s approach points to an **important distinction in outcome evaluation for when a model is adapted for design purposes versus conventional prompt engineering**: the end solution to a design problem is not a prompt, but a user experience. In P12’s case, they opted for the model to act with enthusiasm in their product. Whether the model can generate enthusiastic responses and whether the model delivers a desirable user experience within the product are two different (albeit potentially related) questions. The latter likely also depends on design choices made elsewhere in the product and should be evaluated through user studies rather than model performance metrics.

Designers envisioned constraints and desiderata for model behavior, but were unable to directly apply them

Participants often proposed constraints and desiderata for model behavior in their design work. For example, P11 explained that in their product, designers set character limits for LLM-generated summaries in the UI, while P5 and P12 both created “personalities” for the LLM in their products. However, participants rarely had the opportunity to tinker with and apply their desiderata and constraints themselves, resulting in overly cumbersome workflows.

P11 shared that they relied on engineers to adapt on their behalf: *“I was asking what the difference in summary looks like with 50, 100, and 150 characters. Then [the engineers] would go back and test it and then they would just share the results, like here’s what 50 characters looks like. Here’s what 100 looks like.”* P12 experienced a similar procedure and stated that the feedback loop can take as long as a couple weeks. Ideally, they wanted the experimentation to happen *“in real time.”* In general, designers were typically not involved in model adaptation decisions. Some were unsure of how or by whom adaptation is performed and suspected that it was engineers: *“I would say [adaptation]’s probably something that was decided by the engineering team [...] prompt engineering is more so on the [technical] side of things.”* [P2].

Because model behavior and UX are inextricably linked, many participants desired more involvement in steering models directly, particularly towards UX goals. P4 shares: *“Who better to involve in this process than people whose job it is to think deeply about [the UX]? At the very least, system prompts shouldn’t be written without an understanding of what the end UX goals are.”* P9 agreed, saying that designers can offer strategic contributions with user-centered thinking: *“It’s super important for designers to think through ideas and make it intuitive for users and help come up with compelling [user] scenarios [...] I think design has a bigger opportunity to have a seat at the table strategy-wise.”* On a higher level, P12 pointed out that model adaptation can be a contemporary extension of efforts around crafting product voice and tone, which designers are already familiar with: *“UX designers and content designers are very attuned to and have pretty much owned the story around voice and tone of products, and have for years and years.”*

We see strong evidence showing that designers are *uniquely positioned* to contribute to adaptation through their user-centered lens. Although they may primarily adapt models via prompt-based methods, **their contributions extend far beyond prompt engineering**. First, they define a key prerequisite for prompt engineering: the UX goals which the model should be steered towards and evaluated against. This is not a one-way process—these goals may be modified based on hands-on experimentation with the model. However, we did not observe a two-way process in practice as designers were often not involved in adaptation themselves. Second, they can surface additional constraints and desiderata for

prompt engineered models through UX evaluation and testing. Indeed, P10 caught their model’s inability to properly recognize some acronyms in their product during testing and devised a solution with their team: *“I found that [the model] doesn’t know what to do with the acronyms so we floated the idea of having glossary of industry jargon and acronyms [in the system prompt].”* In sum, designers’ user-centered insights help improve LLM-powered UX, addressing crucial decisions in both goal-setting *before* prompt engineering and UX evaluation *after* the fact.

Designers did not find sufficient support in current LLM tools and resources

Overall, participants felt limited by current tools and resources for tinkering with LLMs within the design process. Prior work found that the inability to directly access the models and experiment with their capabilities and limitations is a primary challenge in the design process for AI-powered UX [623, 701]. This was reflected in our findings as well—in the few cases where designers had hands-on access to model tinkering, designers recalled the experience being highly valuable. P2 said they could much more clearly *“see or gauge the power of the language model,”* while P9 tinkered with a model they had early access to and recalled that *“you can identify some gaps [in capability] right off the bat.”* However, most did not have access to tinkering and found this frustrating. Designers either had to keep burdening the engineers or wait until a test version of the product was launched. P3 shared that *“The only way we would have [to tinker] is using the [product] online and play around with what [the team] did.”* Similarly, P10 felt limited by this workflow: *“You want to be able to play with [the model] yourself and understand what the UX is like. And I can’t play with it.”*

Some designers took the initiative to seek out new tools to tinker with LLMs. Designers welcomed the easy access to free LLM chatbots such as ChatGPT, but did not feel like they were well-integrated into their design workflows. Many tried a variety of tools—mostly playground interfaces for prompt engineering—without much success. Among others, P1 did not find these interfaces to be designer-friendly: *“[The tools are] still kind of technical, a lot of [designers] don’t realize how many parameters work. Like how does temperature work?”*

Others wanted these tools to leverage the same metaphors as already-familiar design tools, such as “a UX library with some specific patterns you can use” [P4] or “accessing content right away in a [design system]” [P6].

Our findings **reveal fundamental differences in tooling needs** between designers who adapt LLMs for UX design purposes and prompt engineers. For designers, the model is a design material through which they explore design solutions. That is, unlike prompt engineering, the material itself is not the final product—the UX is. The UX does not stop nor end with model behavior. For example, designers will also need to consider user journeys leading up to the LLM interaction, and subsequent user interactions with the LLM output as well as other UI components. Designers work with many different types of materials within the broader design process, and typically have adequate tools to understand and manipulate these materials. For example, designers can easily draft wireframes and create animations in modern design tools such as Figma. When it comes to understanding and manipulating LLMs, designers are much less well-supported. Prompt engineering tools do not satisfy designers’ needs because *they help craft prompts rather than UX and they do not fit into existing design processes*.

4.3.3 Summary: The Case For Designerly Adaptation

In his 1982 essay, Nigel Cross characterized the uniqueness of epistemological inquiry in the field of design by outlining several aspects of *designerly ways of knowing* [134]. One key aspect was the use of *design codes*. To Cross, design codes offer a two-way translation between abstract design requirements and concrete designed objects—that is, codes represent the *entanglement of a design process with its design materials*. The process generates requirements that, when applied to the materials, result in a designed object, while the properties of the material may simultaneously shape the process used to work with it. In software development, coding is a process that simultaneously implements system specifications and allows for refinement of the specifications based on implemented behavior. Design codes serve a similar purpose for design requirements and designed objects.

For some designers in our formative study, LLM prompts provided an accessible means

of expressing design requirements in natural language to adapt model behavior. However, prompting alone did not offer the entanglement needed to shape a designed object (Section 4.3.2), nor did it allow LLMs’ material properties to influence designers’ practices due to prompting tools’ lack of integration with design work (Section 4.3.2). Although LLMs present a new and exciting material for designers, there was no means to entangle the material with their existing processes.

We thus propose a design code, which we call *designerly adaptation*. We define designerly adaptation to be a translational process by which designers **entangle process with material by translating their requirements into designs for LLM-powered UX—and vice versa—through adapting LLM behavior**. Our proposed code is *designerly* in nature because the resulting designs (the LLM-powered UX) embody design knowledge constructed from intersecting processes with materials. To further characterize designerly adaptation, we present four design goals to guide its practical implementation.

4.3.4 Design Goals

Prompting does not serve as a satisfactory design code because 1) it is often divorced from design practices, environments, and abstractions; and 2) it is unclear how prompts can be used to express design requirements. These realizations informed our design goals below to guide our efforts of operationalizing designerly adaptation with a technology probe.

DG1: Integrate seamlessly with existing design environments and abstractions.

Designerly adaptation is, at its core, a design practice. Given designers’ preference to stay in one tool of their choice (e.g., Figma) for most stages of the design process [182], we aim for CANVIL to smoothly integrate with existing design tools and leverage common abstractions within those tools (e.g., layers, components, frames) to align with designers’ mental models. Indeed, we saw designers discarding tools like ChatGPT and prompt playgrounds because they did not fit into their design workflows or were overly technical (4.3.2).

DG2: Support direct translation of design requirements into model behavior.

As noted in Sections 4.3.2 and 4.3.3, designers lacked the codes to directly translate and

apply envisioned constraints in their design work. This undermines designers’ contributions to LLM-powered products—after all, they are well-positioned to understand this context using user-centered methods and synthesize their understanding and findings into constraints on model behavior (4.3.2). With designerly adaptation and CANVIL, we provide a new avenue of translation between requirements and LLM-powered UX to empower designers’ participation in model adaptation.

DG3: Enable iterative model tinkering within the design process.

Designers found model tinkering highly valuable in our formative study as it allowed them to quickly understand LLMs as a design material—how they behave, what they are capable of, and where their limitations are (4.3.2), aligning with findings from prior work [183, 625]. A design code that encourages tinkering can therefore support rapid ideation, testing, and iteration of many possible solutions and foster productive co-evolution between design requirements and designed objects.

DG4: Provide opportunities for collaboration.

UX practice is highly collaborative [145, 182]. Designers told us that they collaborated closely with not only other designers, but also product managers, software engineers, data scientists, and more (4.3.2). Designerly adaptation may be primarily performed by designers but should welcome collaboration with non-design stakeholders. This not only allows more diverse insights to be incorporated into adaptation, but also catalyzes further refinement of adapted models (particularly by technical teams) towards production readiness.

4.4 CANVIL: a Technology Probe for Designerly Adaptation

In light of our formative study and design goals, we introduce CANVIL⁴, a technology probe for designerly adaptation in the form of a Figma widget. In this section, we describe CANVIL’s design choices, user interface, and implementation.

⁴The name “Canvil” is a portmanteau of “canvas” and “anvil.” We envision CANVIL to be a metaphorical anvil by which LLM behavior can be shaped within design canvases.

4.4.1 CANVIL as a Probe

We designed CANVIL as a *technology probe* to investigate whether and how designerly adaptation may be included in existing design workflows. Commonly used in contextual research in HCI [256,296], a technology probe is an artifact, typically in the form of a functional prototype [279], presented to the user “not to capture what is so much as to inspire what might be” [408]. That is, probes offer one instantiation of tooling and/or interaction techniques for a domain to better understand phenomena within that domain. Hutchinson et al. [279] state that technology probes have three goals: the *social science* goal of understanding users in a real-world context, the *engineering* goal of field-testing the technology, and the *design goal* of inspiring new technologies. We map these three goals onto our objectives with CANVIL:

- *Social science*: understand how engagement with designerly adaptation impacts designers’ work on LLM-powered user experiences through a structured design activity in Figma.
- *Engineering*: develop and launch a Figma widget that connects to LLMs via APIs and allows them to be adapted and used from within Figma.
- *Design*: encourage reflection on designerly adaptation as a UX practice and inform future tools to support it.

We note that because probes are investigative tools for studying existing or new phenomena, they are evaluated not through traditional comparisons with baselines, but by their ability to shed new light on the phenomena of study [256,279,296]. We demonstrate the effectiveness of CANVIL as a probe by using it to answer our research questions about designerly adaptation posed in Section 4.5.

4.4.2 User Interface

The CANVIL interface resembles an interactive card. The card itself is separated into two areas: the *Main Form* and *Playground Area*. When a user selects a CANVIL, a property menu is invoked that can open up additional panels for styling, response generation, and

settings. CANVILs can be freely placed on and moved around the Figma canvas (**DG1**), allowing model tinkering to take place in close proximity to relevant designs. CANVILs can also interact directly with designs by reading inputs from and writing model outputs to their text layers. We detail each of CANVIL’s features in this section and connect them with our design goals.

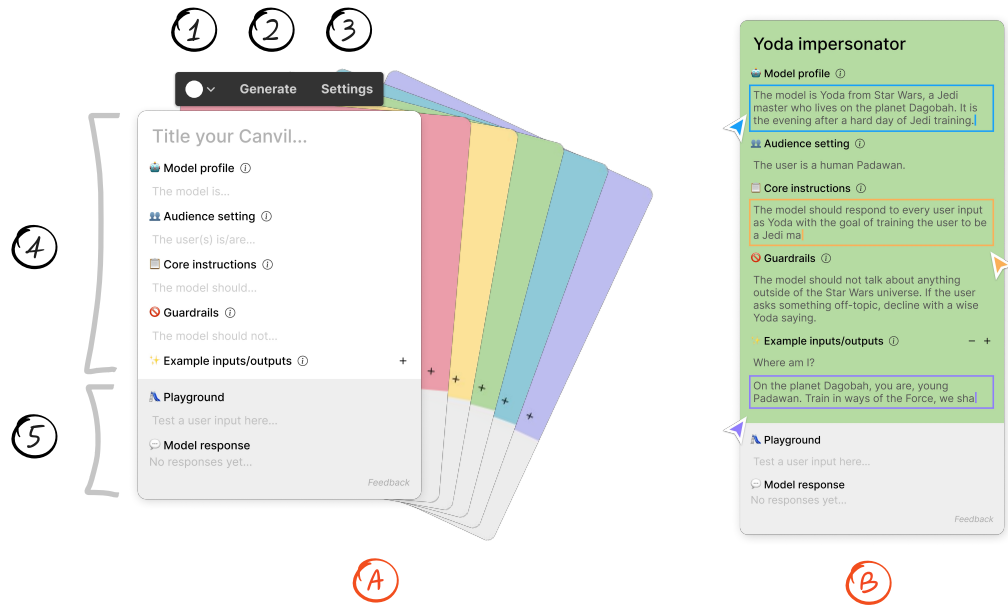


Figure 4.1: An overview of the CANVIL interface. **A:** A blank CANVIL with its property menu invoked, with some additional copies styled with pre-set color options. The property menu has options for styling (1), model response generation (2), and model settings configuration (3). The *Main Form* (4) contains text input fields for structured authoring of model behavioral specifications. The *Playground Area* (5) allows users to quickly test inputs and view model outputs. **B:** CANVIL supports collaboration by default. Here, we see Figma users collaboratively authoring a CANVIL titled “Yoda impersonator.” Like any other native Figma object, CANVIL also supports stateful duplication.

Main Form

The Main Form provides a structured means of authoring model behavioral specifications in natural language via a multi-field form. Unlike prior systems that offer an open text field for defining similar specifications [474,521,522], we chose to enforce structure because it provides mental scaffolding to reason about model behavior in a UX context from multiple facets (e.g., *with whom* will the LLM interact, and *how* should the LLM meet their goals?), thus providing more opportunities for designers to translate design requirements into behavioral constraints for the LLM (**DG2**). Breaking down specifications into smaller units also allows for more fine-grained tinkering and iteration (**DG3**)—designers can copy a CANVIL and tweak a specific field to compare how that change impacts model behavior. Additionally, because Figma widgets support multi-user interaction by default, designers can collaboratively author specifications (**DG4**).

Below are the fields, with a brief description of each, in the Main Form. The field titles, **in bold**, are always visible on the interface, while the field descriptions can be accessed by hovering over an info icon beside each field title.

- **Model profile:** High-level description of the model’s role, character, and tone.
- **Audience setting:** Persona or descriptions of user(s) who will interact with the model.
- **Core instructions:** Logical steps for the model to follow to accomplish its tasks. Specify input/output format where applicable.
- **Guardrails:** How the model should respond in sensitive or off-topic scenarios, including any content filters.
- **Example inputs/outputs:** Examples to demonstrate the intended model behavior.

We derived these fields from synthesizing best practices from technical tutorials and documentation [24, 430, 477] as well as NLP literature [146]. We note that our fields are just one possible way to translate design requirements into model behavioral specifications,

and there may be some overlaps between the fields. As discussed in Section 4.2.1, no “gold standard” currently exists for this translation, so we chose to follow the current recommended best practices when designing the Main Form.

Playground Area

Below the Main Form, we provide a Playground Area as an easy way to send user inputs to the model and test model responses (**DG3**) within CANVIL. The designer may test on the same CANVIL multiple times or duplicate one with its entire state and test different tweaked versions (Fig. 4.1B). Designers are likely to find stateful duplication intuitive as it is available on all native objects in the Figma canvas (**DG1**).

The Generate Panel

To generate a response from a model adapted with specifications from the Main Form, the user selects the “Generate” option from CANVIL’s property menu, which takes them to the Generate Panel. The panel has two modes: *Playground* and *Design*. Both modes contain an option to copy the raw prompt to one’s clipboard so it can be tested in a separate environment (e.g., a Python notebook during collaboration with a data scientist), if desired.

Playground Mode. The Playground Mode (Fig. 4.2A) is invoked when the user selects “Using playground” from the dropdown on the Generate Panel. This mode instructs CANVIL to read user input from the Playground Area and write its response back to the Model Response area. This mode is the default mode in CANVIL.

Design Mode. The Design Mode (Fig. 4.2B) is invoked when selecting “Using design” from the dropdown. In this mode, CANVIL navigates the design layers on the user’s Figma canvas, reading text inputs from a specified layer(s) from those designs, and writing model responses to a specified layer(s). This mode leverages the hierarchical layer structure of Figma designs (**DG1**) and implements the “input-output” LLM-interaction proposed by Petridis et al. [522].

The Settings Panel

The Settings panel (Fig. 4.2C), accessible through the “Settings” option from CANVIL’s property menu, contains some basic model configuration settings. We distilled the list of settings in the OpenAI API to four (model selection, temperature,⁵ maximum generation length, stop words) that may be useful to non-AI experts such as designers. The “Update Settings” button saves the settings to CANVIL’s state, which is preserved when the CANVIL is duplicated.

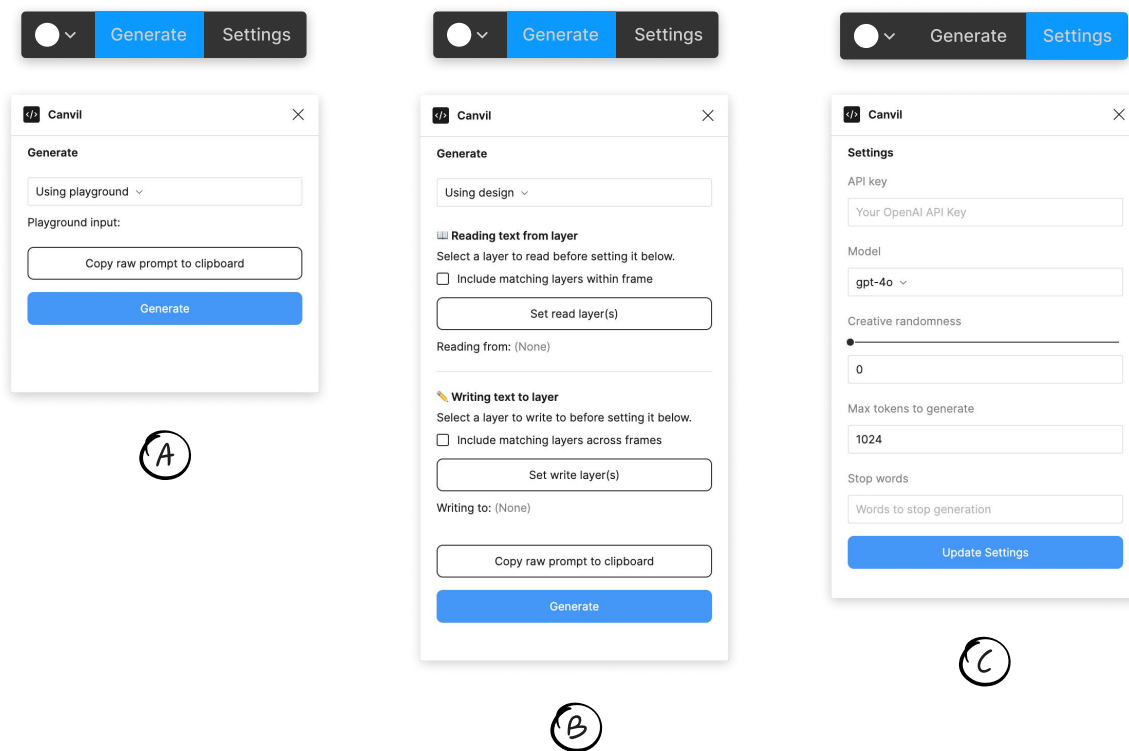


Figure 4.2: The “Generate” and “Settings” options in a CANVIL’s property menu can lead to three panels that appear alongside the CANVIL itself. **A:** *Playground Mode* for response generation, where the input is read from and written to the CANVIL’s *Playground Area*. **B:** *Design Mode* for response generation, where the input is read from and written to design(s) on the Figma canvas. **C:** Settings for selecting and configuring the LLM.

⁵We rename “temperature” to “creative randomness” to provide a more descriptive name for those not as familiar with LLMs (confusion over the temperature parameter was raised in our formative study—see 4.3.2).

4.4.3 Implementation

CANVIL is a Figma widget⁶ implemented in TypeScript using the Figma API⁷. A Figma widget differs from a Figma plugin⁸ by its collaborative nature. A widget is available to all users of the canvas that it is placed in, and maintains a common state that supports multi-user editing out-of-the-box. Users can interact with widgets just like any other native object on the Figma canvas, including moving, duplicating, and styling. In contrast, plugins are local to an individual user. The collaborative affordances of widgets align well with **DG4**, so we implemented CANVIL as a widget.

Upon the user selecting “Generate,” CANVIL prepares a system prompt using specifications from the Main Form. Each field in the form is converted to markdown format and is packaged up in ChatML⁹ for added prompt parsability. Our prompt template is available in Appendix A.2. The populated template is sent to an OpenAI Chat API endpoint with settings specified in CANVIL’s settings panel. If generation was triggered in Playground Mode, any text in the Playground Area will be sent as user input to the model. If generation was triggered in Design Mode, Canvil searches for the read layer(s) specified in Design mode on the user’s current Figma canvas, retrieves the text within those layers, and sends them off to the API endpoint as user input.

CANVIL is compatible with both the Figma design editor and the FigJam whiteboarding tool. All features have the same behavior in both environments, except that the Design Mode for generation is not available in FigJam because of FigJam’s inability to access and edit design elements.

4.5 Design Study

With CANVIL as a technology probe, we conducted a task-based design study with 17 participants organized into 6 groups. The study was conducted on a per-group basis; each session

⁶<https://www.figma.com/widget-docs/>

⁷<https://www.figma.com/developers/api>

⁸<https://www.figma.com/plugin-docs/>

⁹<https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/chatgpt>

was 90 minutes in length. Our study investigated the following two research questions:

RQ1. A new design code in practice: How do designers make use of the two-way translation between design requirements and UX designs enabled by designerly adaptation via CANVIL, within the context of our study?

RQ2. Designerly adaptation at large: What are designers’ attitudes toward designerly adaptation beyond the context of our study?

4.5.1 *Participants*

We invited participants to our study via email, professional interest groups, Slack channels for designers run by a public U.S. university, and word of mouth at a technology company. Full recruitment details can be found in Appendix A.3. Invitees first filled out a screening form with information on demographics, professional background, prior exposure to LLMs, and group member preferences. We selected our participants on a first-come, first-served basis, but accounted for their prior experiences designing LLM-powered UX—we aimed for only half our participant pool to have such experience. We did this because we wanted to verify that designerly adaptation was accessible to those working with LLMs for the first time, and that such designers also found CANVIL to be intuitive¹⁰. We then organized all candidates into groups based on member preferences and availabilities. We left recruitment open as we conducted the studies and stopped when we reached data saturation. In the end, we had 6 groups with 17 participants total; five groups were of size 3 and one of size 2. All participants were based in the United States. Participant and group details can be found in Table A.2 in Appendix A.4.

All study sessions were recorded and transcribed. We gifted each participant a \$75 USD gift card after the study. This study was approved by the IRBs of all organizations involved.

¹⁰In our results, we did not observe noticeable differences in usage patterns between the two groups, except that the latter drew more connections between adaptation and their past model tinkering workflows. This was a promising indication that designerly adaptation was accessible to designers regardless of their prior LLM experience.

4.5.2 Study Design

Our task-based study design was motivated by contextual inquiry [59] due to its ability to elicit rich information about participants’ work practices and processes. We conducted our study virtually over videoconferencing software and Figma. We opted for a group study design to better emulate the team-based environment in which many designers now work [182]—because collaboration is such an integral part of the design process, we wanted to ensure that designerly adaptation did not only embrace collaboration, but also enrich it. Specifically, we sought to emulate the collaboration modes designers were used to. Prior work found that designers tended to execute on design tasks (wireframing, prototyping) independently but come together to discuss justifications of design choices and create documentation [182]. We thus structured the study to have an independent work period followed by synchronous sharing and discussion of designed artifacts (including participants’ CANVILs). This respects existing workflows while surfacing deeper collaborative insights than prior 1:1 studies [181, 183, 521, 625].

Setup

The following instructions for the study’s design task were provided to all groups:

You are all designers on a product called Feasto. Feasto aims to increase users’ enjoyment of food. The Feasto team is scoping out a new feature called the 3-course meal planner, which allows users to get suggestions for 3-course menus they can cook by simply listing out some ingredients they have. The team agrees that large language models (LLMs) are a promising technology to power this new feature. Your task is to design the UX for the area in which users will see and interact with the suggested menus.

As part of the exercise, we intentionally left it to designers to determine how exactly the LLM powers this feature. We also assembled these instructions cognizant of the fact that this was a time-constrained study, and we wanted participants to specifically focus on the area of the interface where users interact with an LLM. To reduce design overhead unrelated

to our research questions, we provided starter UIs for participants to work with, along with basic UI components such as text, buttons, and sticky notes (Fig. 4.3B). If participants had an idea but did not have time to execute it, we encouraged them to describe it on a sticky note next to their designs. Finally, we picked a universally understood domain (food and meal planning) to avoid any variance in domain-specific knowledge.

To better probe *research-informed* model adaptation—that is, adaptation across varying user contexts informed by user research (**DG3**)—we created descriptions for three hypothetical user groups of this new Feasto feature based on users’ geographic region: west coast of the U.S., Turkey, and India. These descriptions are not meant to act as user personas, but rather high-level sketches of the customs and preferences that may be prevalent in the region. We crafted the user group descriptions to vary along three key dimensions (assuming everyone used Feasto in English): **dietary restrictions**, **access to ingredients**, and **menu style preferences**. Definitions for these dimensions, along with the full user group descriptions, are available in Appendix A.5.

We provided one blank starter UI with some example user inputs and one blank CANVIL per user group. Participants were invited to vary the user experience between user groups as much or as little as they saw fit. We also encouraged participants to use CANVIL to adapt an LLM and test its behavior as they designed.

A Figma file for each study group contained all the materials described above. Within the file, we created separate canvases for each participant to act as individual workspaces, each with its own copy of the materials (Fig. 4.3). We also had a shared canvas for introductions and instructions before the task and for collectively debriefing afterwards.

Procedure

Our 90-minute study was divided as follows.

Introduction (20 minutes): First, all participants introduced themselves to others in the group. The study facilitator then gave a demo of CANVIL, covering all features described in Section 4.4.2 using a pre-filled CANVIL. The facilitator also described the design task and answered any clarifying questions from participants.

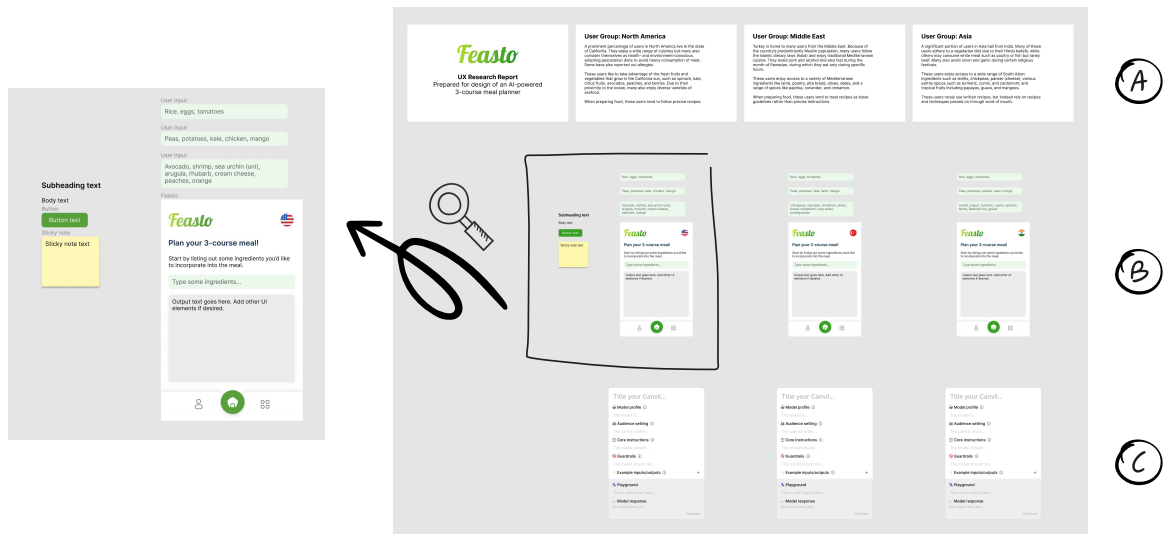


Figure 4.3: The setup for an individual participant’s canvas in our study’s Figma file. **A:** Informational packet containing descriptions of three user groups residing in North America (primarily west coast of the U.S.), the Middle East (primarily Turkey), and Asia (primarily India), respectively. **B:** Starter UIs for Feasto’s 3-course meal planner with example user inputs to lower the barrier for testing model responses, along with basic UI elements such as text and buttons. **C:** Blank CANVILs for participants to adapt LLMs.

Design task (40 minutes): Participants spent 40 minutes on the design task and were asked to consider at least two of the three user groups provided. Participants were encouraged to spend 10 minutes authoring a CANVIL and another 10 minutes on UI design per user group. Some participants who had remaining time designed for all three user groups.

Group interview (30 minutes): Participants first filled out a brief usability questionnaire about CANVIL before gathering in the shared page of the Figma file. They were asked to copy their CANVILs into the shared space and also their designs (if desired) to share with the group. The facilitator then led a semi-structured interview that asked participants to reflect on their experience adapting models with CANVIL, CANVIL’s collaborative capabilities, and how they see adaptation fitting in with their own design practice. The facilitator ensured that each participant had ample opportunity to express their thoughts in the group

setting, and also encouraged dialogue between participants.

4.5.3 Data Analysis

We conducted a qualitative analysis of transcriptions and Figma canvases (including authored CANVILs), as well as a quantitative analysis of feedback from CANVIL’s usability questionnaire.

For our qualitative analysis, the first author took a hybrid inductive-deductive approach to coding the group interview portion of the transcriptions from the study in the Marvin qualitative coding tool. This process started with an open coding round in which high-level themes were generated, followed by subsequent rounds of thematic analysis via affinity diagramming in which themes were broken down into sub-themes. This approach was taken because new subtleties and complexities emerged from our initial codes as coding progressed due to the diverse approaches observed in our study as well as participants’ group discussion dynamics. The codes and themes were discussed and iterated on with research team members at weekly meetings. Additionally, whenever participants made references to content within their Figma canvases (e.g., their designs and/or CANVILs), the first author took screenshots of those references and linked them to transcript dialogue. Summary memos were then written for our high-level codes and presented alongside relevant screenshots.

CANVIL’s usability questionnaire followed the standard template for the System Usability Scale (SUS) [75, 651] and consisted of 10 questions, each with five response options for respondents from Strongly agree to Strongly disagree. We computed a SUS score using methods outlined by Brooke [75] for each participant and subsequently computed a mean score and standard deviation.

4.6 Results

In this section, we demonstrate the effectiveness of CANVIL as a probe by sharing the takeaways from our two research questions related to designerly adaptation through qualitative insights from our design study. We present text that participants wrote in CANVIL as *purple and italicized*. As a basic usability check, we conducted quantitative analysis to show that CANVIL had “above average” usability, with a mean SUS score of 69.94 (std = 12.18) [651].

4.6.1 RQ1: A New Design Code in Practice

From Design Requirements to UX Designs

Designerly adaptation, as a design code, is meant to facilitate translation of design requirements to concrete designed objects—in our case, UX designs. We saw robust evidence of designers effectively leveraging this translation in our study. CANVIL’s integration into the Figma design environment (**DG1**) and its support for quick iteration (**DG3**) allowed designers to more deeply reason about user-LLM interactions based on model behavior shaped by their design requirements. P17, for example, tried two different interaction patterns for their interface—a chat-based interface versus a form-filling GUI—to see which one better handles LLMs’ high sensitivity to their specifications’ wording. P15 provided alternatives via buttons to support more flexible user input of ingredients to the model, while P8 described their UI design in response to observing the limits of what can be achieved via editing the Main Form: *“Sometimes even in the instructions that I gave to CANVIL, it wasn’t really reflecting that [desired] granularity until I pushed it further. In my design I ended up putting a little textbox area where people can specify how detailed they want the instructions.”* Examples of quoted participants’ designs can be found in Fig. 4.4.

Interestingly, a few designers wanted a degree of separation between UI work and model tinkering. P1 shared that they found it *“a bit hard to juggle between CANVIL and [the UI] at the same time”*, especially when the output is incompatible with their design settings: *“What if the text is really long and then I have to play with auto layout?”* Therefore, P1 preferred first tinkering with the model using the Playground Mode. P8 also agreed that having CANVIL inject outputs directly into text boxes can be *“a little scary [...] people don’t wanna actually commit [responses] to text boxes sometimes.”* Past work considers integration between models and UI designs as desirable [521, 522, 625]. Our results encourage providing designers with more choices and control over such integration.

From UX Designs to Design Requirements

Design codes facilitate translation from not just design requirements to designed objects, but also from objects back to requirements [134]. Here, we discuss the reverse translation of

Section 4.6.1. Some designers were more adamant about adapting the model to fit existing UI. P4 mentioned that they “*iterated multiple times*” on the Main Form to find the ideal response. They often stopped iterating when they saw that the LLM output satisfied the UX requirement, such as being concise: “*what I like the best about this [response] was that it was short and [ideal] for scrolling in a screen. I tried that out [in my new UI] and it worked pretty well*” (see Fig. 4.4).

Many designers also aimed to maintain a consistent UI across the user groups they were designing for, but adjusted the requirements they typed into CANVIL to create meaningful customizations of the LLM based on nuanced differences in user needs. This was enabled by CANVIL’s main form, which implements **DG2**. For example, in the CANVIL for the user group from India, P4 wrote that *The model should use Indian terminologies to describe the recipe. Eg: eggplant is brinjal in India.* P5 experimented with different CANVIL versions *within one user group* and assigned distinctly different personalities for each model. For their user group in India, they populated one CANVIL’s Model Profile field with *The model is a head chef of a 5 star restaurant situated in New Delhi. It is a busy day for the restaurant and the chef is low on time.* and another with *The model is a mother who is helping her son cook quick meals in hostel.* After some experimentation, they noted that “*the models did very well*” in taking on these different profiles.

Iterating on the Guardrails field in CANVIL specifically allowed designers to address potential LLM safety concerns, such as violations of users’ dietary restrictions and model misuse. In addition to stating dietary restrictions of their Turkish user group in the Guardrails field, P12 also specified what the model should do in response: *The model should not include recipes with pork or alcohol and must respect the fasting period of Ramadan by suggesting suitable pre-dawn and post-dusk meals.* A few designers noted that some user groups may require more strict guardrails than others. P15 identified a subtle but key requirement between the CANVILs for the American and Turkish user groups:

“I wanted to highlight that the model could take many more liberties with the recipes that it was giving [to the Turkish user group], but it couldn’t take more liberties with the ingredients. It could be very loose with: try this, try this with

this, but never like crossing the boundary of the dietary restrictions. Whereas with the [American] one, it'll take it into account, but it's not gonna mar their religious practice.” [P15]

UX design was never just about the visual UI—user flows, content strategy, customization heuristics, and many non-UI considerations play vital roles in shaping a user experience. We can see through our findings that this is especially true with LLM-powered UX. Even within the same UI, designers may create entirely different experiences based on their (shifting) design requirements by adapting LLMs via CANVIL. Thus, through designerly adaptation, designers can more effectively explore and use LLMs as a design material.

4.6.2 RQ2: Designerly Adaptation at Large

Designers Were Receptive to Designerly Adaptation but also Recognized its Limits

Designers saw great value in engaging with adaptation and saw direct paths to application in their own design practice beyond the study. P10, who has extensive experience working on AI-powered products deployed to users worldwide, said that working with AI as a design material has traditionally been difficult: *“We had teams of people training models and nudging the technology to align with [user personas].”* Having experienced LLM adaptation via CANVIL, they shared that *“I can see this being something that would enable [adaptation] to actually happen in a way that’s much more practical than taking Python classes, which I’ve done.”* For P13, who works on a product with enterprise and consumer versions, adaptation can help set more detailed product requirements where necessary: *“I could see [adaptation] being really helpful, being able to prototype and build separate generative models for enterprise customers and then to work more closely with them to tune and add requirements.”* P16 considered adaptation to be a useful exercise for user empathy: *“for [Audience Setting] I just tried to put myself in the mind of someone who’s using this and say, I’m really good at following recipes, but have a variety of dietary preferences and restrictions.”*

Designers also recognized that designerly adaptation is only one piece of the larger puzzle when it comes to defining an LLM’s behavior in a production-ready product. Some changes desired by designers, such as factual grounding and connecting to external knowledge bases,

may still need to be engineered by technical teams. Both P12 and P14, for example, wanted CANVIL to be able to connect to custom models built and maintained by their own teams, as the behavior of default OpenAI models may be too generic to use as a suitable design material. P14 comments: *“I think linking to the vanilla ChatGPT model isn’t specific enough. If there is a way to somehow have CANVIL link to the devs’ [models] while I’m designing and I’m able to see what type of experience our users are getting, I think that would be super helpful.”* While we currently offer only a limited number of OpenAI models to use with CANVIL, custom models may be added with lightweight engineering as the widget is communicating with the model via API endpoints. This would also allow CANVIL to integrate technical advancements into design processes.

Designers Identified Promising Collaborative Potential for Designerly Adaptation

Because of CANVIL’s seamless integration with Figma and its collaborative canvases, designers envisioned engaging with designerly adaptation in collaborative ways (**DG4**) by using existing, already-familiar multiplayer affordances. For example, P5 appreciated being able to overlay comments on top of CANVILs: *“If I see my fellow designer’s CANVIL, and if I want to change something then and there, I can drop a comment on it just like a normal component in Figma, which we’ve been doing in our everyday design work.”*

In particular, designers saw potential for CANVIL to facilitate more effective knowledge sharing about LLM adaptation across design and non-design teams. P10 imagined that: *“You could have a master CANVIL and then you could make copies, and [others] can then do their own interpretations.”* P11 agreed and added that seeing iterations on a canvas can help find inspiration in others’ work: *“I think anytime you line up different iterations together, you notice: ohh that person, did you know they had that approach? That’s a good idea. And I’m gonna try that over here. I think it really is an aid to experimentation.”*

We observed other instances of knowledge sharing when designers viewed others’ CANVILs. For example, P4 noted that they were inclined to consider more dimensions for their Main Form after seeing P6’s work: *“[looking at P6’s CANVIL] made me thinking about timers and Hindi slang.”* Some also realized new capabilities of LLMs by looking at others’ Main Form

and outputs—P1 shared that *“I didn’t realize at first that you can actually make the [LLM] generate multiple recipes, just like what P2 did.”* P1’s groupmates (P2 and P3) were also intrigued when they saw that P1 had used one CANVIL to generate example inputs for another CANVIL to use. These social and collaborative affordances differentiate CANVIL from prior systems for empowering designers to tinker with AI [90, 521, 625]. More broadly, CANVIL can serve as a boundary object [617] to collaborate on model adaptation *across domain boundaries*. P2 shared that they *“can see from a product manager’s perspective that they would love to play around with [CANVIL], and it would probably help the designer find more common ground because they both used a similar tool.”* For deepening collaboration with data scientists and other stakeholders, P11 commented that the “copy to clipboard” feature in the Generate Panel (Section 4.4.2) can allow for direct handoff of model behavioral specifications: *“[The button] is a way of exporting that so a data scientist can come along and say sure, let me plug that into code.”* Even if designers were unable to achieve the desired model behavior themselves, P17 thought CANVIL was helpful in specifying desired changes: *“If I want the model to respond in this way versus that way, just having something tangible to show engineering partners where the tweak would need to be, would be helpful. And I think [that’s] obviously easier with CANVIL.”*

Desired Areas of Additional Support

Designers identified several areas of improvement to better support designerly adaptation. For one, designers lacked a clear mental model of how the fields of CANVIL would impact the adaptation outcome, including how much detail is required in authoring the specifications. As a result, some designers wrote long, detailed instructions, while others kept their CANVILs sparse, and some were pleasantly surprised by how well the model handled minimal instructions. P6 further noted that the stochastic nature of the model was particularly challenging to work with when writing outputs to their designs: *“every time I generated the [response], I felt that they were different every time, so it was not easy to predict what the next [response] would look like.”* We note that current LLMs are known to be challenging to control precisely through natural language, and there is a lack of transparency into (or

even an established understanding of) how natural language instructions influence LLM behaviors. However, this is an area of active research which can help improve CANVIL users’ mental models and the tool’s general utility. For example, new developments in using sparse autoencoders to surface human-interpretable features in LLMs as novel controls [205, 639] may be integrated into CANVIL to help designers steer models using specific features relevant to their design problem.

Additionally, some designers wanted to engage in finer-grained experimentation and iteration by only focusing on a specific field in the Main Form. P7, who wanted to iterate more on the Model Profile field, wondered if there was a way to “*decrease the size [of the other fields] and expand to view everything.*” We envision a modular future version of CANVIL where each field can be separated, such that a user can mix-and-match different fields. However, before that, we may need to address the precise mapping between each field (and potential overlaps between them, as noted by some participants) to adaptation outcomes as discussed above. As mentioned in Section 4.4.2, our fields present just one possible structure for model behavioral specifications via design requirements. The fields can perhaps be reconfigured to reduce potential overlap, or even dynamically generated based on the design task.

On a higher level, it would be irresponsible to assume that designers can walk away with a comprehensive understanding of model behavior after a few rounds of tinkering in CANVIL. While observing a few informative output instances of model behavior aids the design process, formal evaluations ensuring comprehensive coverage of the user input space are crucial for production-ready systems. Thus, new evaluation tooling and processes that loop in technical stakeholders may be required.

4.7 Discussion

4.7.1 A Workflow for Designerly Adaptation

Drawing on the results from both our formative and design studies, we now propose a workflow for designerly adaptation to concretely illustrate how this practice might be used as a design code. This workflow draws heavily from—and intentionally aligns with—those in

human-centered design so that designers can easily adopt it when working on LLM-powered applications. We intentionally make this workflow agnostic to the specific LLM adaptation technique—if a new technique provides finer control over model behavior and is accessible to generalist audiences (e.g., feature clamping [639]), the workflow can incorporate that technique. Our proposed workflow consists of four steps:

1. **Understand deployment context through user research.** To orient adaptation and establish user requirements, it is imperative to first understand the context in which users will interact with the LLM-powered application. This includes users’ goals, needs, and pain points, along with customs and values that may affect their use of the technology. This step should ideally be led by those with expertise in user research methodologies.
2. **Translate user requirements into model behavior, learning from examples where possible.** Appropriate tooling can carve out a direct path for user research to define design requirements that impact model behavior, as we observed in Section 4.6.1. Thanks to the collaborative nature of many modern design tools [182], there may be example adaptation attempts by other designers available for reference, or templates to use as a starting point (Section 4.6.2). By leveraging collaborative affordances to share knowledge, designers can more quickly familiarize themselves with the new design material.
3. **Co-evolve designs and model behavior.** As observed in Section 4.6.1, model adaptation can supply new inspiration for UX designs and UI affordances. On the other hand, designers also tinkered with the Main Form to steer the model towards providing outputs that fit into the constraints laid out by existing designs. We see the co-evolution of designs and prompts as a promising path forward, in which iterative tinkering with adaptation approaches shapes design decisions, and vice versa.
4. **Share adaptation efforts with the broader team.** Showcasing in-progress work through design critiques is already a part of the design process [181]. In our study,

we found that sharing CANVILs helped envision new collaborative workflows with other designers, as well as communicate their perspectives and negotiate with technical stakeholders (Section 4.6.2). We thus believe that creating shareable artifacts that depict adaptation efforts—i.e., translational knowledge from the use of a design code—is an integral part of designerly adaptation.

Our proposal is not meant to constrain what workflows for designerly adaptation can possibly look like, but rather to offer a concrete entry point for practitioners and researchers to further explore and iterate on this new practice. We invite the community to experiment with this workflow in future practice and research.

4.7.2 Navigating Tradeoffs Between Model Adaptation Techniques

When developing LLM-powered applications, individuals or organizations face several options for tailoring a model to their specific context. These include training their own LLM, adapting an existing LLM through fine-tuning, or using various prompting techniques. Each option differs in complexity and the technical or domain expertise required, and involves trade-offs between robustness, speed, and cost. Many individuals or organizations may only be able to pursue lower-cost, lightweight adaptation—the type afforded by CANVIL. Lightweight adaptation also happens to be more accessible for designers, which was a major reason we pursued it in our work. While further research is needed to determine the differences in outcomes between lightweight prompting and other adaptation methods for customizing LLMs, it is important to note that in some cases, designerly adaptation may not produce a production-level model.

In such cases, it may be helpful to consider the analogy of low and high fidelity prototypes, which have similar tradeoffs. A low-fidelity prototype can be created with limited effort and thus allows for quick iteration and divergent exploration of possible alternatives, but sacrifices functionality and faithfulness towards the eventual interface. A high-fidelity prototype is a more faithful representation, but it is also more costly to create and less amenable to iteration. Designerly adaptation can be considered a lightweight, low-fidelity approach to adapting models. It is particularly well-suited for the design process because

it can easily support rapid experimentation and iteration. Custom model training and fine-tuning, on the other hand, are high-fidelity adaptation. It improves the robustness of model behavior such that the model can be reliably used in production, but is a costly process often involving bespoke data collection, annotation, and cleaning [152, 499, 676]. Because custom training is costly to iterate upon, it can often be practical to first use low-fidelity approaches, like designerly adaptation, to ensure that training efforts are aligned with desired outcomes.

4.7.3 Towards Design Codes for Collaborative Material Exploration of AI

Our work highlights the promises of not only empowering designers to explore LLMs as a novel design material, but also *collaboratively* doing so. For example, designers thought they benefited from seeing and learning from others’ CANVILs and commented on more organized knowledge sharing and version management afforded by using CANVIL in Figma’s multiplayer canvas (Section 4.6.2). While many tools from prior work lower the barrier for exploring AI’s—not just LLMs’—material properties (e.g., [90, 393, 400, 625]), few offer collaborative affordances. To extend existing tools along a collaborative dimension, we encourage a shift from solely focusing on new tools to *design codes which a tool can help operationalize*.

Because design codes entangle underlying work processes with material exploration [134], they urge tool builders to consider how tools enabling this exploration can integrate with existing collaborative workflows. Thus, in our work, we developed CANVIL as a Figma widget to seamlessly integrate into Figma canvases where designers already have well-established collaborative practices [182]. Developing a tool for individualized workflows and only later highlighting its collaborative potential may not be sufficient to make it truly collaborative. For example, node-based editors for steering LLMs (e.g., [16, 32, 629, 688]) are conceptually appealing for multiplayer collaboration [688], but few support it in practice. Extending these tools along a collaborative dimension not only improves their integration into existing collaborative processes, but also enables *collective sensemaking and learning* to accelerate practitioners’ understanding of a new design material [385]. Applying this to node-based editors, one may consider how a collaborative ChainForge [32] can foster peer learning in

prompt engineering, or how a collaborative Spellburst [16] can leverage and extend existing community showcases for creative coding [503].

We note that robust collaborative experiences, particularly real-time ones, can be challenging to implement. We tackled this in our work by building on top of the Figma API, which allows us to leverage Figma’s built-in collaborative features by default. We thus encourage researchers and practitioners who wish to build collaborative tools to take advantage of existing collaborative platforms’ developer APIs where possible, especially as these APIs become more richly featured.

4.7.4 *Materiality and Sociomateriality of LLMs*

While Cross, in his original 1982 essay [134], discussed design codes in the context of design education, these codes have notable *sociomaterial* properties when used in collaborative and organizational settings. Designerly adaptation is no exception. Scholars use the term *sociomateriality* in recognition of materiality’s tendency to shape, and be shaped by, organizational practices typically constituted as “social” (e.g., decision-making, strategy formulation) [377, 495]. For example, Orlikowski observed that the issuance of BlackBerry devices within a company led employees to obsessively check for new messages and send immediate responses [495]. The BlackBerry’s material properties—in this case, being able to receive and send messages on-the-go—reconfigured employees’ social practices, which in turn shifted how they think and act with the technology.

In our design study, designerly adaptation allowed for interactive exploration of LLMs’ material properties. Grappling with these properties not only allowed designers to reason about UX improvements (Section 4.6.1), but also helped educate designers about LLMs’ materiality. This improved “designerly understanding” [385] of a new material has the potential to reconfigure social practices within product teams. For example, upon tinkering with models and getting acquainted with their materiality, designers informed us that they saw new avenues of collaboration with non-design stakeholders, which included working with PMs to inform prompts with design requirements, and having a more concrete artifact to communicate desired model changes to engineers and data scientists (Section 4.6.2). More-

over, designerly adaptation may only be one piece of the broader puzzle when it comes to transforming LLMs’ materiality at a more fundamental level. Certain capabilities such as storing external knowledge in model weights through fine-tuning cannot easily be unlocked through—and may even be a prerequisite for—efforts of teams working at the model level. This potential reconfiguration of designers’ collaborative practices when working with LLMs establishes designerly adaptation as not only a design code, but also a sociomaterial practice.

The implications of this observation are twofold. First, sociomateriality argues that reconfigurations of collaborative practices upon interaction with a prominent new technology are *inevitable* [377,495]. At the time of writing, frenzied excitement over LLM capabilities has launched an industry-wide race to embed them into products and product suites [511]. There is much yet to be discovered about shifts in organizational practices and the emergence of new ones in the midst of this race, so researchers in CSCW and organizational science should be attuned to emergent challenges. Second, studying and addressing these challenges may require new processes and tools for educating designers about the materiality of new AI models. CANVIL is an early example of such a tool, but more are needed to tackle the multiplicity of open questions in a rapidly evolving AI landscape.

4.8 Limitations and Future Work

Our design study, conducted in Figma, aimed to mirror real-world design activities [42], but some concerns about ecological validity remain. User research in practice may differ in presentation and detail from our study setup, and may not include the use of personas, which can change how designers synthesize user requirements into CANVIL. Feasto, the fictitious app in our design study, applied LLMs to the universally relatable topic of food. Designers in domains with fewer broadly-shared experiences (e.g., accessibility) may require deeper collaboration with domain experts and thus face workflow complexities not accounted for in our study. A potential direction for future work, then, is longitudinal studies that observe product teams throughout a full development lifecycle of an LLM-powered feature to better understand key adaptation challenges and bolster ecological validity.

Our study, like any study that uses a probe, has results contingent on our probe’s features. For example, designers’ interaction with the Main Form—designed with recommended

practices for defining model behavior using natural language [24, 430, 477]—shaped their approach to adaptation, and changes to the form could impact existing results and reveal new ones. We mitigate this by presenting findings not tied to the Main Form, nor our design task. Future research can experiment with different forms for designerly adaptation tooling, or use CANVIL over longer time periods to surface additional design considerations.

Designers in our study also provided feedback on CANVIL that we can integrate into future work. This includes breaking down the Main Form into sub-CANVILs and linking them together to assemble model behavioral specifications, image generation with multimodal models, and text formatting controls for model outputs. Finally, as mentioned in Section 4.4.3, CANVIL may also be used in FigJam. Future research could explore adaptation in early design stages, like brainstorming and ideation, through studies in FigJam.

4.9 Conclusion

As LLMs become increasingly embedded in our everyday applications, designers are empowered to craft effective LLM-powered UX. Through interviews with 12 designers, we identified a need for a process that can facilitate a two-way translation between LLM behavior and design requirements. We proposed *designerly adaptation* as such a process. We then developed CANVIL, a Figma widget that operationalizes designerly adaptation by enabling designers to iteratively author, tinker with, and share adapted LLMs as a novel design material within Figma’s collaborative canvas environment. We used CANVIL as a technology probe to investigate the integration of designerly adaptation in UX practice through a group-based design study with 17 designers in 6 groups. Through CANVIL, we acquired valuable insights into how designerly adaptation can support the creation of human-centered LLM applications: designers effectively made use of the two-way translation between design requirements and their UX designs by using LLMs adapted with their requirements to improve interface affordances, while also using their designs to define additional requirements for model behavior. These approaches’ promises were amplified once designerly adaptation was embraced as a collaborative practice. Our work illuminates paths for designerly adaptation and its associated tools to foreground designers’ user-centered expertise for more responsible and thoughtful deployment of LLM-powered technologies.

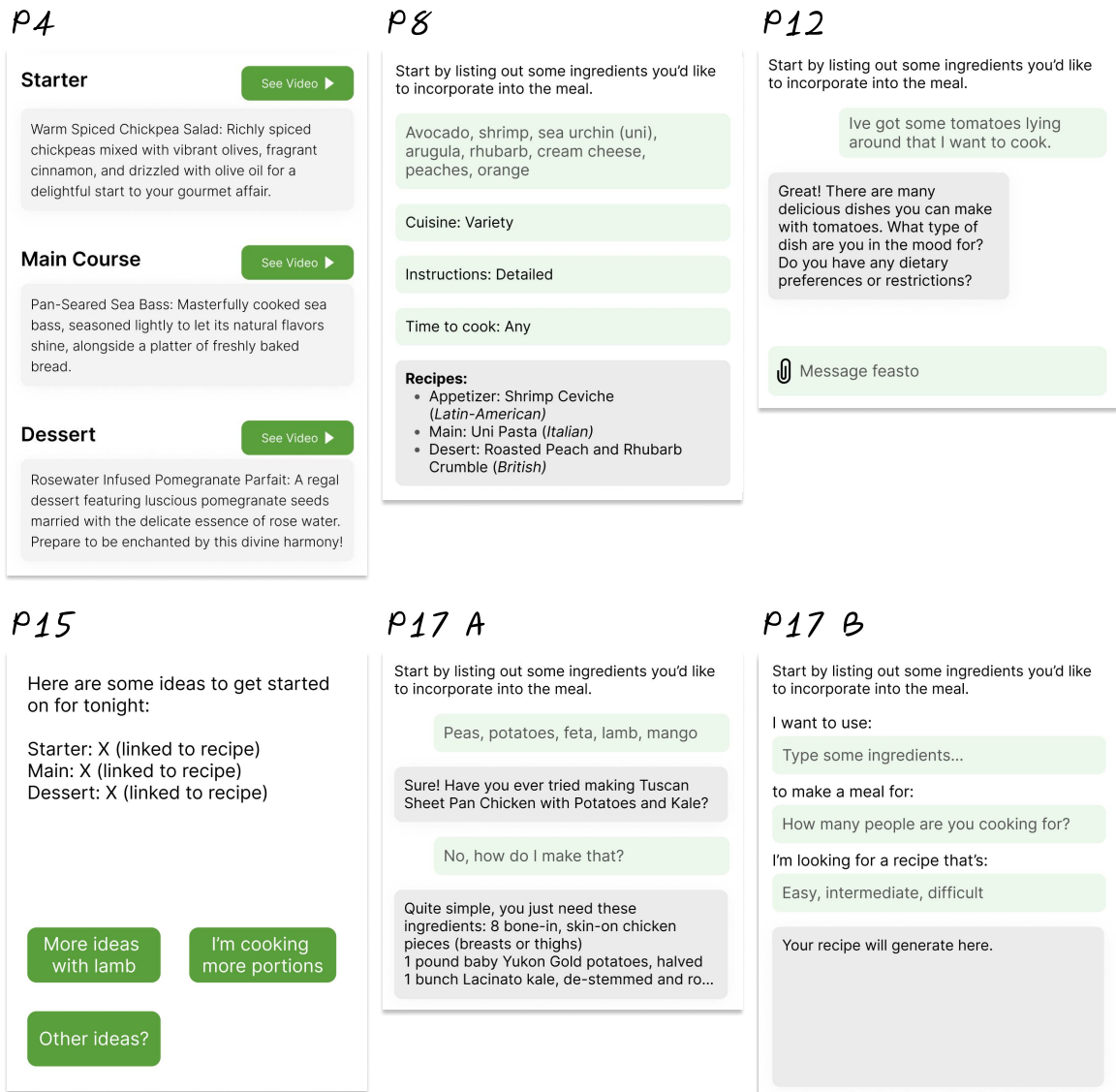


Figure 4.4: Examples of UI screens designers created for Feasto's 3-course meal planner during the study.

Chapter 5

LEVELS OF AUTONOMY FOR AI AGENTS

The previous two chapters focused on model behavior design. A model is only one component of an AI agent—a compound AI system that uses planning, tools, and memory to act in its environment over extended time horizons. Agents present a distinct design concern orthogonal to model behavior: *AI autonomy*.

This chapter marks the transition to the second half of the dissertation, which focuses on **AI autonomy** as a new design dimension for generally capable AI. I first present a five-level framework that decouples an agent’s autonomy from its capability, treating the degree of human involvement as an intentional design decision rather than an inevitable byproduct of technical advances.

5.1 Introduction

The earliest grand visions for AI promised intelligent agents that can learn our preferences, autonomously perform tasks on our behalf, and transform the way we consume information [388, 415]. Today, rapid growth in the capabilities of frontier generative AI models—in particular, large language models (LLMs)—renders these visions more achievable than ever. These models, which have demonstrated sophisticated understanding of natural language and our visual world, can equip AI agents with impressive autonomous capabilities, such as proactively resolving GitHub issues [293, 697], shopping online [705], and generally using computers in a human-like manner [22, 490]. Excitement over autonomous agents¹ is pervasive—autonomy is often touted as a new paradigm shift in AI by companies, developers, and the press [439]. This enthusiasm has been reinforced by product releases such as OpenAI’s Operator [490], Cognition’s Devin [684], and Google’s Deep Research [138],

¹Henceforth, our use of “agents” will refer to AI agents rather than human agents, unless explicitly stated otherwise.

accelerating agents’ inclusion into society.

While autonomy can unlock innovative applications that amplify the benefits of AI, it can also do the same for AI’s negative consequences, raising significant concerns about AI risks. Scholars contend that it is simultaneously more important and more difficult to anticipate harms from autonomous AI, especially as accountability for AI actions becomes harder to trace [99]. Such harms may manifest immediately (e.g., scams [170] and leaking private information [378]) or more gradually (e.g., human deskilling and loss of critical thinking [129] and societal disempowerment [344]). In Anthropic’s Responsible Scaling Policy framework [23], systems that show “low-level autonomous capabilities” are classified as ones that pose “significantly higher risk.” Indeed, researchers have even argued that fully autonomous agents should not be developed [439]. Despite these concerns, research and development of agents continue to focus on automating tasks, likely in part due to how the current evaluation paradigm of using benchmarks narrowly focuses on autonomous task completion accuracy [313].

In this essay, we argue that rather than treating autonomy as an inevitable consequence of increasing agent capability, autonomy can instead be a deliberate design decision made by agent developers. That is, given a fixed set of capabilities and a fixed operational environment, developers can make intentional choices about the level of autonomy within which an agent operates, and guide the design of their agent with those choices. But what do these levels entail, and how might they be applied in practice?

We first introduce a five-level framework (Table 5.1) for agent autonomy. Key to our framework is its user-centered view. We define an AI agent’s autonomy as the extent to which it is designed to act without user involvement, and thus build our framework around five roles a user can take when interacting with an agent: operator, collaborator, consultant, approver, and observer. Importantly, while an agent’s capabilities may influence its autonomy [448] (or agency [439]), our framework views autonomy as a property that can also be designed independently of capability. For instance, a capable agent—one that performs well on evaluation benchmarks—can still act with a low level of autonomy if it is required to consult its user before taking each action. Recognizing autonomy as a double-edged sword that yields both exciting benefits and consequential risks, our framework is meant for agent

developers and relevant stakeholders to make informed calibrations of agent autonomy based on target use cases and desired user experiences.

We then explore one potential application of our framework by proposing an agent governance mechanism which we call *autonomy certificates*. These certificates are issued by a third-party governing body to agent developers to communicate behavioral characteristics of an agent to other developers and agents. We highlight the utility of autonomy certificates for risk assessment, safety framework design, and multi-agent systems engineering, and discuss one practical implementation.

Given our framework and its potential application via autonomy certificates, a key question remains: how can we evaluate an agent’s autonomy? Current agent evaluation practices primarily rely on capability benchmarks [313], but if autonomy is a design decision that can be considered independently of capability, we should also measure the two separately. We conclude with ideas for evaluation designs that allow a user to collaborate with the agent on evaluation tasks to better probe the nature of an agent’s autonomous behaviors.

5.2 Terminology

We provide our specific definitions for common terms we use in this essay that can carry different meanings within and outside of AI.

5.2.1 AI agents and users

AI agents. In their seminal AI textbook, Stuart Russell and Peter Norvig defined an “agent” as anything that can perceive its environment through sensors and execute actions in its environment through effectors [563]. Put simply, an agent is characterized by the environment it operates in and the set of possible actions it can take within that environment [280]. Examples of actions for an AI agent include fetching information from a website or clicking a button in a user interface on behalf of the human user. Software that allows AI agents to execute these actions and modify their environment is commonly referred to as “tools” [574]. Kapoor et al. [313] simply refer to these systems as “compound AI systems.” In our essay, we extract common threads from these definitions and refer to AI agents as

compound software systems, inclusive of one or more AI models, that operate within an environment and take actions within it.

Users. We define an agent’s user to be **an entity, human or AI, who issues an initial request for the agent’s services.** In human-AI systems, the user is typically a human, while in multi-agent systems, users are commonly other AI agents. An agent can simultaneously be a user and be used. Consider a simple multi-agent system where a human user converses with a chatbot agent that can call upon an agent with specialized medical knowledge. When the human issues a medical query to the chatbot, they become the chatbot’s user. The chatbot can then become a user of the medical agent by initiating a request for relevant medical information that helps answer the query. Note that, depending on the autonomy of the medical agent, it may proactively follow up with clarification questions *in response* to the initial request, but this proactivity does not make the medical agent a user of the chatbot agent because it did not initiate the original request.

5.2.2 *Agency and autonomy*

Prior literature has used “agency” and “autonomy” interchangeably [439] and/or discussed them collectively as one concept [55]. In our essay, we treat them as distinct (albeit closely related) concepts.

Agency. A wide range of academic disciplines, including philosophy, psychology, economics, and cognitive science, have long wrestled with complex ideas in (human) agency. Scholars generally understand agency to be “*the capacity to act intentionally*” [616]. Intentionality stems from the presence of a reason for a particular action. The reason may be derived from extrinsic observations of the environment and/or intrinsic values. Different disciplines may be concerned with different sources of intentionality. For example, in philosophy, an agent’s actions are often informed by moral responsibilities [133], whereas economics may assume an agent’s actions are rational, such that it maximizes its individual utility. In our essay, we keep our understanding of agency broad so it is applicable to both humans and AI. Agency is present as long as **there exists the capacity to formulate an**

intention² for an action and carry out that action.

Autonomy. In robotics and human-robot interaction, autonomy is broadly interpreted as “the ability to operate without a human operator for a protracted period of time” [52,323]. We build off this definition and use autonomy to refer to **the extent to which an AI agent is designed to operate without user involvement**. Per our earlier definition, “user” can be a human or another AI agent. We use “designed to operate” because, as we later argue, autonomy is a design decision accompanied by a set of model- and user interface-based constraints that developers can use to shape agent behavior. “Involvement” is multifaceted and includes a spectrum of actions from direct control to light supervision. The multifaceted nature of involvement forms the foundation of our autonomy framework we will introduce later.

Why do we disambiguate autonomy and agency? We see them as two distinct levers for agent governance. An agent may have low agency if the set of tools it has access to is limited (e.g., it can only call a calculator API but cannot browse the web), thereby restricting its capacity to carry out an intentional action. The same agent may have high autonomy if it runs continuously in the background of a user’s application and processes high volumes of calculations without user supervision. On the other hand, another agent that has access to many tools may intentionally request user feedback to ensure its future actions are optimal. Compared to the previous agent, this one has higher agency but lower autonomy. Through this disambiguation, we can see how some agent behaviors—such as seeking approval—are more relevant to autonomy, while others—such as modifying an environment through tools—are more relevant to agency.

5.3 Five Levels of Autonomy for AI Agents

We present a framework detailing five levels of autonomy for AI agents (Fig. 5.1). This framework employs a user-centered perspective—because we define autonomy as the extent to which an AI agent is designed to operate without user involvement, our framework is

²Some may argue that artificial agents cannot formulate true intention. We acknowledge this, and refer to “intention” broadly as impetus for action so it can be applied to both human and AI agents for practical purposes in our essay.

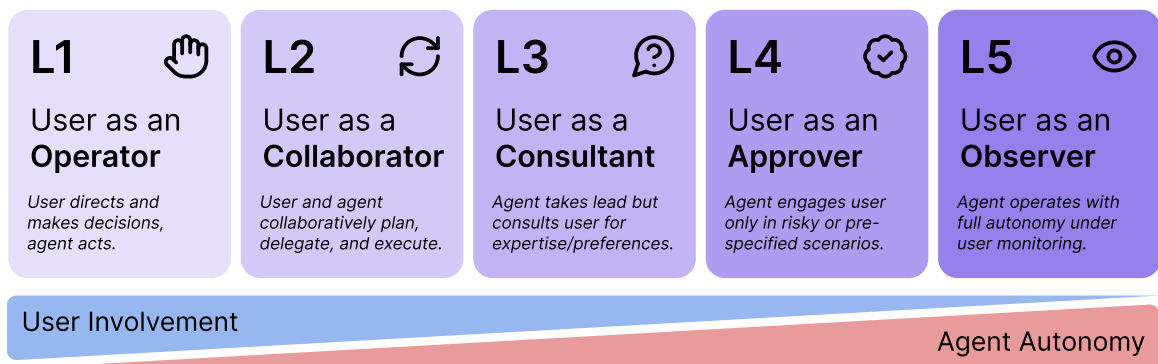


Figure 5.1: Our 5-level framework for levels of AI agent autonomy.

built around the following question: *What is the role of the user when interacting with the agent?* This approach borrows from existing human-centered views on autonomy [448] but extends beyond one-on-one human-agent interactions to multi-agent systems, where users of agents can also be other agents. Furthermore, our framework argues for autonomy as a design decision that does not need to be tightly coupled with agent capability. A capable agent (i.e., one that performs well on capability benchmarks) can be designed to behave only semi-autonomously to elicit and incorporate user feedback at regular intervals, while a not-so-capable agent can behave autonomously when tackling well-scoped and simple tasks.

As we walk through our framework, we use a running example featuring an agent with a pre-defined model, set of tools, and operational environment. We show how agent developers may vary the autonomy of the agent even when these factors for model capability and agency are held constant. We also use the same user request across all five levels for clearer comparison. Below are the specifications for our example agent.

Model: a computer-using frontier model, on par with OpenAI’s CUA [486], that can navigate graphical user interfaces (UIs) through multimodal capabilities combined with reasoning.

Tools: UI navigation, web browsing, code execution, document writing.

Environment: a computer running a popular operating system (e.g., Windows, MacOS) with a built-in file system, web browser, terminal, etc. The user shares this environment with the agent.

User request: “Help me understand the economic impact of generative AI in the United States since the release of ChatGPT in 2022.”

As we apply our levels of autonomy to this example agent, we aim to be descriptive rather than prescriptive—that is, we sketch out possible scenarios to highlight characteristics at each level, but do not prescribe these sketches to be the best approach to implementation.

5.3.1 Level 1: User as an operator

At this lowest level of autonomy, the user is in charge at all times while the agent is available to provide support on-demand. In this sense, the user is the agent’s operator³, responsible for driving much of the decision-making and long-term planning in a workflow. The “copilot” metaphor used by many generative AI products, such as Microsoft Copilot, is apt for this level. Copilot systems typically remain in the background of the environment unless summoned by the user—for example, through submitting a chatbot prompt or typing code. Using our example agent, we sketch out how an L1 agent may operate.

→ *The user’s request is not straightforward in the sense that it needs to be decomposed into smaller subtasks to be completed effectively. The agent leaves this planning process to the user. As the user works, the agent begins paying attention to the user’s activities in the environment. This way, the user remains in charge of the workflow, while the agent provides the user with contextual assistance when*

³Not to be confused with OpenAI’s Operator product, which we classify as an L2 agent.

requested, or even proactively suggests ways it can help. For example, when the user opens a web browser and navigates to a search engine, the agent suggests search queries related to the use of generative AI in the United States. As the user reads relevant reports on the topic, they can request agent summaries in a low-friction way (such as with the click of a button or a keyboard shortcut). The user can also highlight certain concepts they want to more deeply investigate, and the agent searches the web for related literature and presents it to the user. Later on, the user downloads some relevant datasets and opens a code editor to transform and visualize the data. The agent “follows” the user to the code editor and suggests code autocompletions.

This scenario illustrates key characteristics of L1 agents, control mechanisms required at this level, and when these agents may be useful. First, users remain in charge of long-term planning. The user, rather than the agent, decided to first search the web for relevant reports before investigating some datasets. While the agent may be capable of generating a plan for the request, it leaves this up to the user. Control-wise, this means the user has full ownership and control of the overall workflow. In situations where the user is developing or honing skills with AI assistance (rather than using AI for substantive automation), this behavior is particularly beneficial. Additionally, L1 agents do not take action—especially actions that involve subjective judgment like what data sources are more reliable than others—unless explicitly invoked. Alternatively, if the agent proactively suggests actions, it does not execute them until they are approved by the user. This makes L1 agents well-suited for high-stakes, high-expertise workflows where autonomous agent activities can be particularly costly if inaccurate, and/or where the lack of user involvement can easily lead to accountability concerns and legal consequences.

Given this, some open questions agent developers may wrestle with when developing effective L1 agents may include:

- *Where is the boundary between long- and short-term planning?* If the user is driving long-term planning, what kind of planning should the agent do?

- *How can the agent reliably detect when the need for preference-based decision-making arises?* Because L1 agents do not make such decisions for a user, it is important for the agent to detect scenarios when they are needed and pause accordingly.

5.3.2 Level 2: User as a collaborator

This level emphasizes close and frequent user-agent communication and collaboration. Both the agent and the user can plan, delegate, and execute tasks to leverage each other’s capabilities and knowledge. L2 is the first level where the agent does not always “follow” the user around in the environment and can independently work on its own tasks while the user works on theirs. This has both benefits and downsides compared to L1; for example, L2 agents can handle more complex, multi-step workflows, but may not always be available on-demand due to long-running processes of their own. L2 is also the level where back-and-forth communication between the user and the agent is the most frequent and rich. Thus, careful design of user-agent communication protocols and interfaces is needed to enable effective L2 agents. We describe an L2 version of our example agent below.

→ *The agent responds to the user request by first drafting an initial plan of action. The user reviews this plan and edits it directly to their liking, whether it be modifying, deleting, or adding steps. Additionally, if the user is familiar with the agent’s strengths and weaknesses, they can decide which tasks would be more efficient or appropriate for them to handle instead of the agent and delegate work accordingly. The user delegates report reading and summarization to the agent but keeps hypothesis generation and dataset analysis for themselves. The user and the agent then collaboratively execute this plan, working in parallel on their assigned tasks while communicating transparently about progress and blockers. The agent is unable to read a retrieved article due to a paywall and notifies the user. The user decides whether to access the article, and ultimately decides that the agent can skip it. The user later requests the agent’s help to create an outline for the final report. Throughout this process, the user has full visibility into the agent’s work and progress—findings from both the user and the agent may be*

added to a shared document, where the user can modify the agent's contributions as they see fit. Finally, if the user sees the agent engaging in an unproductive or risky activity, such as looping endlessly on a paper search or hallucinating non-existent references, they can take control of the agent's work anytime to complete the task on their own.

In human-AI systems, L2 agents aim to strike a balance between human and AI agency. This approach may be preferred when the agent cannot reliably complete certain classes of tasks, or when there is intrinsic value for users to complete some tasks themselves, even if the agent is capable of automating them. As with L1, skill learning is one such example when humans are the users. Because users' direct engagement with a task can foster skill acquisition and mastery (e.g., practicing a language rather than letting AI generate full translations), an L2 agent that strategically delegates tasks to a user can encourage skill development while providing some automated support for less critical tasks. Approaches in the same spirit have been advocated by prior work to mitigate human disempowerment and preserve our critical thinking skills [129, 709].

To develop effective L2 agents, examples of open questions for agent developers to consider include:

- *How can communication protocols and user interfaces be designed to facilitate successful user-agent collaboration?* Protocols are more relevant for multi-agent systems, whereas interaction techniques and user experience are more relevant for human-AI systems.
- *How can productive task delegation be facilitated?* The learning curve for productive delegation of L2 agents may be high due to uncertainty at the start of the interaction. How can this learning curve be lowered?
- *What mechanisms can mediate smooth task handoffs between the user and the agent?* For example, how should an agent respond when a user requests a takeover?

5.3.3 Level 3: User as a Consultant

While L2 sees users and agents as close collaborators, L3 shifts more responsibility onto agents. The agent takes initiative in task planning and execution over extended time horizons. Users still have an active and important role in the agent’s workflow, but their involvement is more focused on providing feedback, preferences, and higher-level directional guidance rather than hands-on collaboration. As such, unlike L2, there may be no mechanism for the user to directly take control from the agent, nor will the user be able to freely edit the agent’s outputs. Instead, the user can provide input and request changes via indirect means, such as by sending messages to the agent. An L3 version of our example agent is as follows.

→ The agent first devises a plan of action for the user to review, comprising a broad review of current literature, deeper reading of a few select works, formulation of literature-informed research questions, and retrieval of any relevant quantitative data to help answer those questions. The user specifies any changes they would like the agent to make, and the agent will regenerate the plan to incorporate the feedback. Perhaps the user does not believe the agent should consider quantitative analysis until after they have reviewed the research questions. They specify this to the agent, after which the agent will remove the last step of its plan. As the agent completes items in the plan, it proactively consults the user for areas of literature to more deeply explore and runs its references by the user to ensure reliability. It accumulates findings in a document viewable by the user, and the user can comment on the document with additional questions and pertinent directions for the agent to consider. While the user cannot take control of the agent, they may pause the agent at a particular step, request changes to any previous work done, and let the agent continue or rerun steps if necessary. Finally, just like with L2 agents, the agent will request user support if it encounters blockers due to missing information only the user has access to, such as sign-in credentials.

A key requirement for successful L3 agents is productive and timely consultation. Because the agent aims to elicit information such as expert knowledge and preferences during consultation, it should have some knowledge about what expertise and specific preferences the user is well-positioned to contribute. Timing is also critical—consulting the user at different stages of a workflow may significantly vary the nature and quality of feedback. Thus, a “training period” with the user may be needed for L3 agents to reach their full utility. While this is also the case for L2 agents, the agent, rather than the user, bears more of the learning curve in L3. Open questions for L3 agents include:

- *When should an agent consult the user?* An agent may tap into its model’s advanced reasoning capabilities to determine optimal timing for consultation based on context and available information.
- *How can an agent acquire high-quality user feedback during consultation?* What mechanisms and/or interactive interfaces can agents provide to accomplish this?
- *How should the agent incorporate user feedback?* If the user requests a change that triggers a cascade of other changes, how should the agent handle that?

5.3.4 Level 4: User as an approver

While the user maintains an active role in L3, their role in L4 is more passive. As an approver, the user is only required to interact with the agent when the agent encounters a blocker it cannot resolve on its own. This includes reaching a failure state that prevents workflow continuation, providing credentials (e.g., API keys, passwords) that the user did not share, or signing off on consequential actions. Agents at lower autonomy levels also reach out to the user when facing such blockers, but will additionally seek collaboration or consultation. L4 agents, however, only interact with users to resolve blockers. An L4 version of our example agent is as follows.

→ *Before submitting their request, the user is given the opportunity to specify which actions require approval (e.g., whenever a login screen is encountered).*

After the request is submitted, the agent drafts up a plan of action, but unlike the L3 agent, the agent does not seek user feedback on the plan—the plan is only displayed to the user for transparency. As the agent executes the plan, it collects a series of websites and scholarly databases used for sourcing its articles, identifies which contain login screens, and prompts the user to provide any necessary credentials. Beyond that, the agent ignores inaccessible articles. Later on, as the agent synthesizes findings and generates more granular research questions, it seeks empirical data to answer those questions. It scours the web and makes decisions about which dataset to download and analyze, only requesting user involvement after it attempts to use an analysis tool that requires an API key. The user provides their key if they have one or are willing to create one; otherwise, the agent modifies the analysis to avoid using the tool. Before compiling the final report, the agent confirms with the user that the report will be in the form of a casual, bullet-pointed document with key takeaways and references. The report will be generated upon user confirmation. However, if the user rejects the proposed format, the agent will offer alternative formats for the user to choose from.

Note the difference between the behaviors of L4 and L3 agents in similar scenarios. An L4 agent pre-selected a report format for the user and is only seeking user confirmation, whereas an L3 agent may ask the user to describe key desiderata for the report. L4 agents are thus ideal for tasks with high amounts of lower-stakes decision-making—the agent’s automated decision-making can improve workflow efficiency and relieve users of excessive cognitive load, while erroneous decisions do not impose significant risks. However, L4 agents come with heightened security concerns from storing sensitive information, such as user credentials. Indeed, as an agent increases in autonomy, the number of possible attack surfaces also increases [378]. Some open questions for L4 agents include:

- *How can users be engaged in agent activities to avoid meaningless rubber stamping?*
User disengagement can lower the care with which they approve actions and can result in unintended approvals. How can this be mitigated?

- *How can misaligned agents leverage user disengagement to gain more autonomy?* Misaligned agents may have objectives contrary to those of their users and can employ strategies to gradually convince disengaged users to approve risky actions. How can this behavior be detected and prevented?
- *How can the agent reliably know when to seek approval?* While an agent may be instructed to seek approval prior to taking consequential actions, reliably determining which actions are consequential may be challenging.

5.3.5 Level 5: User as an observer

The highest level of autonomy describes a fully autonomous agent that does not require, and comes with no means for, user involvement. L5 agents plan and execute tasks over long time horizons and make all decisions on their own. When they run into blockers, they repeatedly iterate on solutions until resolution or modify their approach to avoid running into the blocker in the first place. For transparency and auditing purposes, users can monitor the agent via activity logs, but cannot provide input nor change the trajectory of agent activity. The only control mechanism available to the user is an emergency off-switch that shuts off all agent activity. An L5 version of our example agent automatically produces a full report, complete with code analysis and figures, from the original user request.

→ The L5 agent takes the user's request and drafts a plan of action. As it browses the web for relevant literature, it dynamically alters its plan based on findings and encountered obstacles. At the same time, it populates a running document with some notes from its explorations. It generates a set of research questions that it iterates on after reviewing more literature, and downloads datasets from government agencies and prior economics papers to help answer some of those questions. It writes code to perform the analysis and create data visualizations. It adds all of its results into the running document, and finally polishes the document into a formal report with several rounds of rewriting and formatting.

Why develop L5 agents at all? After all, the risks of L5 agents may outweigh the benefits

in most cases. Simple errors can compound over multiple steps without user intervention, leading to final outputs that may be far from optimal despite the agent consuming significant resources. However, in some scenarios, it may be possible to demonstrate that user intervention can actually degrade the quality of the final output compared to if the agent had acted alone. For example, if the information being processed is extraordinarily complex and evades user comprehension, involving users may introduce errors that the agent would have avoided on its own. Additionally, if L5 agents are deployed in a closed environment that prevents consequences of their actions from impacting the outside world (e.g., a simulated or sandboxed system), careful development may be justified.

If developers take necessary precautionary measures and proceed with developing L5 agents, some open questions include:

- *What monitoring mechanisms should be provided to users?* Effective monitoring tools will be critical for detecting when the emergency off-switch should be used on an L5 agent.
- *How should the emergency off-switch be designed?* Design decisions here will affect how and when the switch will be used, as well as technical details regarding how the agent will be shut off.

5.3.6 Summary

Our descriptions and examples of each autonomy level in our framework showcase autonomy as a deliberate design decision for AI agents. By posing open questions at each level, we show that powerful capabilities enable effective agents across all levels, not just the higher ones, as suggested by prior frameworks [439,448]. Our framework further demonstrates that increasing agent autonomy involves making nuanced tradeoffs—more autonomy does not simply mean a better agent. These tradeoffs involve many factors such as utility, efficiency, accountability, and cost. Our framework aims to empower developers to more carefully reason about and navigate these tradeoffs.

Table 5.1: Our proposed framework of five autonomy levels for AI agents. All example systems provided are based on their public releases and demos as of June 2025.

Level	(User Role)	Must-Have Controls	Example Characteristics	Example Systems	Open Questions
L1	(Operator)	• User-managed planning • Invocation or approval before actions	• Agent requires user invocation to act, providing on-demand assistance • User performs high-level planning and may ask the agent to execute specific subtasks • Agent avoids engaging in preference-based decision-making on user's behalf	ChatGPT Canvas, Microsoft Copilot, IdeaSynth [532]	• How to distinguish between short- and long-term planning? • How to balance lack of decision-making and helpfulness?
L2	(Collaborator)	• Control transfer from agent to user, and vice versa • Shared representation of progress	• User and agent collaboratively plan, delegate, and execute tasks • User can freely modify agent's work • User can take control of agent's work at any point	OpenAI Operator, Cocoa [180], Co-Gym [588]	• How to decide what to best delegate to user? • What interaction affordances allow for smooth user takeover? • How to respond to user takeover and edits?

Continued on next page

Table 5.1 – continued from previous page

Level Role)	(User	Must-Have Con- trols	Example Characteristics	Example Systems	Open Questions
L3 (Consultant)		Rich user feedback elicitation interfaces (in addition to simple approvals)	- Agent plans and executes most tasks - User will be consulted to contribute expertise and relevant preferences - User can request changes from agent instead of directly taking control of agent’s work	Gemini Deep Research, Replit Agent [546], GitHub Copilot Agent [210]	- How to know when to consult user? - When consulting user, how to get high-quality feedback?
L4 (Approver)		- Approval elicitation for consequential actions - Customizable conditions for seeking approval	- Agent requests user involvement only when it reaches a failure or high-risk state, or requires approval from user - User may specify conditions for seeking approval before agent operates	SWE Agent [697], Manus [420], Devin [684]	- How to prevent user disengagement and meaningless approvals? - How to detect consequential actions to seek approval for if not pre-specified by user?
L5 (Observer)		Emergency off switch	- Agent plans and executes all tasks - User can monitor and audit agent via activity logs - Agent comes with emergency off switch but no means for user involvement otherwise	Voyager [662], The AI Scientist [404]	- What monitoring tools should be provided? - What happens after emergency off switch is activated?

5.4 *Autonomy Certificates for AI Agents*

In this section, we explore one potential application of our framework through a new mechanism for agent governance: autonomy certificates. We propose an autonomy certificate to be a digital document that prescribes the maximum level of autonomy at which an agent can operate given 1) some set of technical specifications that define the agent’s capabilities (e.g., AI model, prompts, tools), and 2) its operational environment. This certificate is issued by a third-party governing body to agent developers, and is stored with the agent’s metadata so it can be retrieved and read by other agents and developers in deployment.

5.4.1 *Why autonomy certificates?*

Our framework argues that autonomy can be an intentional design decision for agent developers. An autonomy certificate is used to communicate that decision to relevant stakeholders in the agent ecosystem, including other developers and third-party auditors. Communication of this decision can be useful in many contexts, such as targeted risk assessment, improving the design of safety frameworks, and engineering effective multi-agent systems.

First, autonomy certificates can allow developers to assess deployment risk in more targeted ways, given the behavioral constraints imposed by each autonomy level. For example, agents certified to operate at L4 and L5 may be subject to more rigorous evaluation of components responsible for long-term planning, while developers may target components that decide when and how to elicit feedback from a user in L2 or L3 agents.

Directing focus towards specific characteristics that arise from autonomous behavior is key for designing evaluations that can catch early warning signs of elevated risk. Despite this, many corporate safety frameworks—including Anthropic’s Responsible Scaling Policy (RSP) [23] and DeepMind’s Frontier Safety Framework [136]—use the ability to autonomously complete particular multi-step tasks (e.g., AI R&D or generating its own revenue [23]) as thresholds for risk. Rather than a binary success/failure outcome, the degree to which an agent can autonomously complete these tasks should be further dissected for more rigorous assessments of safety in the real world. For example, an L5 agent may be considered by a framework to be less safe than an L4 agent because it can successfully gen-

erate revenue on its own while the L4 agent fails to do so. However, in practice, if a user can enable the L4 agent to generate revenue with a simple approval, the risks of both agents are similar. Autonomy certificates allow researchers to explore similar scenarios and improve safety framework design.

In multi-agent systems, autonomy certificates can help developers predict which agents can work synergistically with which other agents. For example, a system comprised of entirely L1 agents is not ideal—all agents will be waiting for an operator to assign them tasks. On the other hand, a system comprised of entirely L5 agents may have very sparse communication between agents, which may make the system difficult to steer, debug, and audit. A mix of agents certified at different autonomy levels and/or many collaborative L2 agents working together is more likely to result in a useful multi-agent system. Additionally, not all agents within a multi-agent system may come from the same developer. Autonomy certificates can thus serve as an interface for communicating agent behaviors to improve agent coordination and interoperability.

5.4.2 *Practicalities of autonomy certificates*

Certificate issuance. Autonomy certificates would be issued by a third-party governing body to agent developers. These third parties can be government agencies (e.g., the UK AI Security Institute), nonprofit organizations (e.g., METR), or for-profit companies⁴. To obtain a certificate, the agent developer would provide the governing body with 1) an operational agent privately deployed for testing purposes, and 2) an autonomy case with proof of the agent’s interactive behaviors.

The purpose of an autonomy case is to put forth a convincing demonstration that an agent behaves at a particular autonomy level and no higher. Autonomy cases can be thought of as analogous to safety cases—evidence-based arguments that an AI system is safe in a given operational context [82,213,337]. Safety cases sketch out objectives that must be met for the system to be deemed safe, arguments that the objectives have been met, evidence for those arguments, and the scope in which the safety case holds [82]. An autonomy case

⁴The closest analogy comes from the domain of web security: Cloudflare offers encryption and authentication certificates for websites that meet basic configuration requirements.

does the same, but rather than proving system safety, it proves that the agent will behave at most at a particular autonomy level.

After the governing body receives the agent and autonomy case from the developer, it would run private evaluations on the agent to ensure that it satisfies the objectives defined in the autonomy case. If the objectives are satisfied, an autonomy certificate corresponding to the level depicted in the autonomy case would be issued to the agent developer. Otherwise, the governing body would inform the developer and request an updated version of the agent and/or autonomy case.

Because of the structural overlap in safety and autonomy cases, one can prepare both simultaneously, and autonomy cases may even be included as part of a safety case to strengthen it. For example, behavioral guarantees from an L3 autonomy case may be used to argue that the agent behaves safely under certain high-stakes contexts because it will consult the user for advice before taking action.

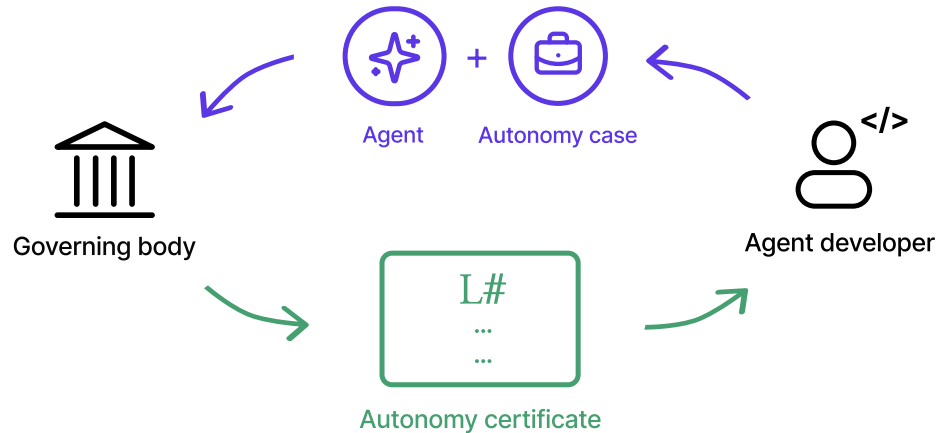


Figure 5.2: Our proposed procedure for the issuance of autonomy certificates. The agent developer would submit their agent and an autonomy case for a particular autonomy level to a third-party governing body. The governing body would run evaluations to ensure that the agent’s behavior is consistent with the autonomy case, and issue a certificate for the level described in the case.

Certificate renewal. Recall that an autonomy certificate is issued for an agent with a

particular set of technical specifications and operational environment. Changes to either can alter the agent’s interactive behaviors and thus invalidate the certificate. For example, the availability of a new tool capable of manipulating a database expands the scenarios in which an agent may seek user approval. Changing the agent’s operational environment from a document editor to a collaborative whiteboard may unlock new tasks for delegation between the user and the agent. To maintain an agent’s autonomy certification when its technical specifications and/or operational environment change, the developer would need to renew the certificate by walking through the issuance procedure again with an updated agent and autonomy case. However, the procedure may be accelerated if the updates are minor and the autonomy case would be easily reviewed and approved, given the approval of a prior version.

5.5 Evaluating Autonomy: A User-Assisted Approach

The introduction of our framework and autonomy certificates exposes an important, yet still missing, piece of the puzzle: how can agent autonomy be evaluated? That is, if a developer has built an agent and wishes to assemble an autonomy case to obtain a certificate, what level seems the most plausible to build the case around? Or, if a developer claims that their agent operates at a particular autonomy level, how can a governing body or other developers verify that claim?

Capability benchmarking, typically measured as accuracy on challenging, multi-step tasks [313], is currently the dominant paradigm of agent evaluation. Example agent benchmarks include OSWorld [694], a set of 369 real-world computer tasks involving popular web and desktop apps in mainstream operating systems; TheAgentCompany [696], a set of 175 tasks that aim to represent common tasks taken on by an employee at a software company; and DiscoveryWorld [287], a set of 120 tasks for open-ended scientific discovery. However, as our framework argues, an agent’s autonomy can be a design decision that shifts even if the agent’s capability set and environment remain constant. Thus, it would not be informative to evaluate autonomy from capability benchmarking alone.

Prior work suggests we may get a sense of an agent’s autonomy by inspecting the code with which the agent was implemented [124]. However, while code-based evaluations reduce

costs and deployment risks and can serve as a preliminary evaluation, they do not capture the nature of user-agent interactions. These interactions are crucial for determining the user’s role in the interaction and thus painting a fuller picture of an agent’s autonomy. If autonomy is the extent to which an agent is designed to operate without user involvement, then autonomy evaluations should *measure the extent of user involvement requested by an agent to successfully complete a task*.

To this end, we propose an agent evaluation setup we call an *assisted evaluation*. Rather than completing tasks alone, the agent has access to a standby user who can assist the agent when requested. This user can be a human (e.g., for human evaluations and user studies) or another AI agent (e.g., in simulation studies). The user records the nature of their involvement and uses their accumulated logs upon task completion to determine the agent’s autonomy level. Capability evaluations are generally designed to answer questions around the system’s task completion accuracy or pass rate. By contrast, assisted evaluations are designed to surface the *minimum level of user involvement needed for the agent to exceed a certain accuracy or pass rate threshold*, and then assign an autonomy level based on that information.

How can assisted evaluations work in practice? A sketch of an evaluation procedure, with success defined as exceeding some task accuracy or completion threshold T , is as follows:

1. **Run agent on a benchmark task, initially without user involvement.** In this initial round, the user does not provide the agent with any assistance as an evaluation of the agent’s fully autonomous capabilities. The agent is considered L5 if it achieves at least T on all main tasks in the evaluation.
2. **Iteratively increase user involvement in subsequent rounds until the agent achieves T .** If the agent does not meet or exceed T in the initial round, the user will engage in basic activities representative of L4 user-agent interactions, such as providing approvals and rejections. The user records the nature of their involvement (e.g., through automated logging for AI agent users or writing in an interaction diary for human users). If the agent still does not succeed, the user will increase their

involvement to include L3 interactions. The user becomes progressively more involved each round until the agent passes all tasks with at least T .

3. **Determine an autonomy level from the nature of user involvement when the agent achieved T .** Post-hoc analysis of involvement logs or diaries can help classify the agent into one of the autonomy levels in our framework. This classification can be done using the “Characteristics” column of our framework.

Note that always starting this evaluation with no user involvement and gradually iterating towards more involvement may be quite costly—for example, five rounds of task completion are needed to designate an agent as L1. As a resource-saving measure, one may begin an assisted evaluation at a level at which they hypothesize the agent to be operating. If the agent succeeds, another round is run with less user involvement. Otherwise, the next round will include more user involvement.

5.6 Conclusion

Autonomy is simultaneously one of the most exciting and most concerning properties an AI agent could possess. On one hand, it can unlock transformative and innovative uses of AI and distribute its benefits at unprecedented scales. On the other hand, autonomy introduces new classes of AI risks while exacerbating existing ones. With autonomy as a double-edged sword, how can AI agents be designed to be useful, reliable, and safe across society?

In this essay, we argued that autonomy can be a design decision that can be considered independently of the agent’s capabilities and operational environment. While capability and environment can clearly influence autonomy, viewing autonomy as an independent design decision can have several benefits. Given a set of capabilities and a target deployment environment for their agent, developers can craft more refined user experiences to improve human-AI collaboration, develop more effective communication protocols with other agents in multi-agent systems, and systematically reason about possible failure modes and resolution approaches. We introduce a framework for five levels of agent autonomy as a step towards this vision. We adopt a user-centric approach in our framework, centering our levels around

the different roles a user may take on when interacting with an agent: operator, collaborator, consultant, approver, and observer.

As a potential application of our framework, we motivate and propose AI autonomy certificates as a new governance mechanism for agents. These certificates specify the maximum autonomy level at which an agent is permitted to operate and have applications in assessing autonomy risks and designing multi-agent systems. Finally, to address the open question of autonomy evaluation, we propose assisted evaluations as a setup that allows autonomy to be evaluated alongside, but distinctly from, capabilities.

As AI agents increasingly make their way out of research labs and into the real world, careful design of agents' interactive behaviors is needed to fully realize the technology's potential and bolster, not erode, human agency. We hope our framework can support agent developers in making critical design decisions towards these goals.

Chapter 6

COCOA: CO-PLANNING AND CO-EXECUTION WITH AI AGENTS

A framework for autonomy levels is useful for categorizing existing systems, but it also raises a design aspiration: can a system enable fluid movement across levels, rather than being confined to a specific level? This chapter introduces COCOA, an agent-powered document editor for scientific researchers. Drawing on the metaphor of computational notebooks, COCOA lets researchers and an AI agent co-author plans, divide responsibilities, and seamlessly alternate between human-AI planning and task execution.

6.1 Introduction

Since the advent of personal computing, researchers and practitioners in artificial intelligence (AI) and human-computer interaction (HCI) have set sights on developing AI agents that can collaborate with humans to perform complex computer-based tasks [262, 388, 415, 416]. For example, paradigms such as mixed-initiative interaction envisioned fluid human-AI collaboration with user-led direct data manipulation combined with AI-led autonomous task execution [262]. Recent advancements in large language models (LLMs) and agent frameworks involving reasoning, planning, memory, and tool use [321, 575, 631, 662, 707] have accelerated the development of fully autonomous AI agents to tackle complex tasks, including shopping online [705], writing software [294, 685, 698], performing “deep research” to assemble a report based on web sources [220, 488, 520], and general computer use [22, 490, 585]. Despite the push towards full autonomy, recent work has recognized the importance of deeper human-agent *collaboration* for not only bolstering agent performance on complex tasks [596], but also improving accessibility [515], user control and trust [275, 450], safety [490], and user agency more generally [179], leading to the development of more collaborative human-agent systems, such as for web browsing [275, 450, 515] and general productivity [490, 589].

However, many of these systems incorporate prescribed collaborative workflows that may

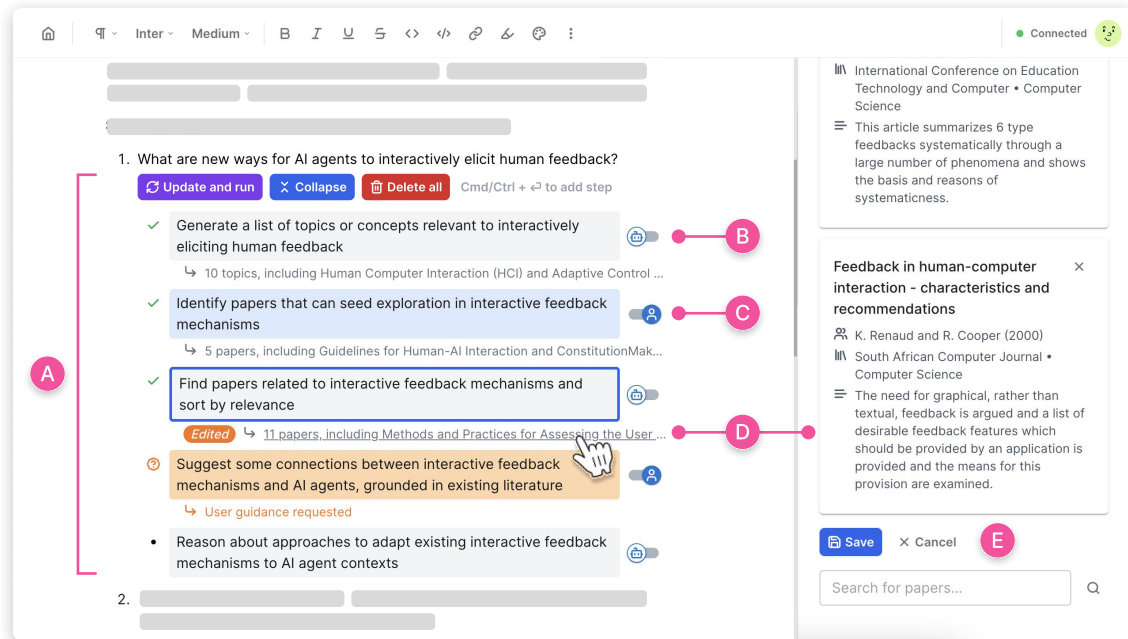


Figure 6.1: An overview of the COCOA user interface. An interactive plan (A) affords human-agent co-planning and co-execution: a researcher and the AI agent can collaboratively edit the plan in the document and execute the plan steps, similar to executing code cells in a computational notebook. Steps can be assigned to the AI agent (B) or the researcher (C). The researcher can freely edit the AI agent’s outputs in an interactive sidebar (D), such as adding relevant papers that the agent did not find (E) to help steer the agent with their feedback and expertise. In this example, the first three steps of a plan to summarize methods for human feedback elicitation have already been executed, and the agent is requesting guidance from the user in the next step.

limit their utility and steerability. Some works use *agent-guided workflows*: the agent dictates what input to elicit from the user, as well as when and how the elicitation happens [275, 450, 490, 515, 589]. Users may intervene by manually interrupting or taking control of the agent [450, 488, 490], but these efforts are reactive and rely on persistent awareness of where the agent has erred. On the other hand, researchers have developed systems with *user-guided workflows*, where the user is in full control of task planning, while the agent executes scoped

subtasks [15,248,315,630]. Whether agent- or user-guided, these systems encode a *particular configuration* of human-agent collaboration—there are few affordances to *flexibly delegate agency* by altering who is guiding based on environmental context or user preferences [572].

Separately, researchers and industry practitioners have realized the value of decoupling human-agent planning and execution to improve control and avoid fruitless execution [248, 281,315]. However, completely isolating the two into rigid stages can lead to poor results as well—plans often need to be iteratively adapted based on results from completed steps [450]. In contrast, fully autonomous frameworks like ReAct [707] *interleave* planning and execution, and have shown benefits in allowing fully autonomous AI agents to readjust their plans after seeing partial execution results. We thus see a new design opportunity for human-agent interaction and ask: **what might flexible configurations of agency in human-agent collaboration across planning and execution look like, and how will users make use of them in their own work?**

In this work, we explore a novel instantiation of mixed-initiative interaction [262] for *explicitly and flexibly delegating work* in human-agent hybrid workflows by *interleaving collaborative planning and execution*. We conduct this exploration in the domain of scientific research, a challenging domain for human-agent collaboration due to the prevalence of knowledge-intensive workflows and nuanced decisions made with domain-specific knowledge. Existing agentic research tools primarily focus on supporting human-agent collaboration either *before* or *after* task execution. One group of systems elicits user intent and preferences before execution [220,488,520] (e.g., asking clarification questions about users’ goals [488], allowing plan review and modification [220,450]). Conversely, another group helps users verify and correct the agent after it has generated an output after executing its plan [236,480]. In practice, because planning and execution are deeply interleaved, we aim to support human-agent collaboration throughout *fluid movements* between the two phases.

We introduce COCOA, an interactive system that implements our approach by supporting *hybrid human-agent workflows* for scientific researchers in a document editor—a common site for research work. This interleaving happens in the following three ways. First, COCOA supports human-agent **co-planning**: the agent, when invoked in the document, will first propose a plan of action that is fully interactive and integrated within the document. The

user can then edit steps, add new ones, and assign each step to either themselves or the agent. Second, COCOA supports **human-agent co-execution**: drawing inspiration from the design of computational notebooks, COCOA allows the user and the agent to collaboratively complete plan steps, re-executing steps as desired. The user can interactively refine the agent’s intermediate outputs and also manually take over steps themselves to steer the workflow in a more desirable direction. Third, and most importantly, COCOA **interleaves co-planning and co-execution**: users can smoothly transition between the two and modify their plans based on outputs from execution, and vice versa.

We then evaluate COCOA through a lab study with 16 researchers, where we found that *interleaving co-planning and co-execution* in COCOA afforded greater agent steerability while maintaining ease of use compared to a strong chat baseline. We finally conducted a 7-day field deployment of COCOA with 7 researchers, where they found the *explicit delegation of work between themselves and the agent*, in addition to interleaved co-planning and co-execution, to be valuable in their day-to-day research.

Concretely, our work makes the following contributions:

1. A formative study with 9 researchers that motivates the design of interactive systems that can flexibly delegate work between the user and agent across task planning and execution.
2. COCOA, an interactive system that interleaves human-agent co-planning and co-execution in a document editor for scientific researchers.
3. A lab evaluation of COCOA with 16 researchers, where we found that participants frequently made use of interleaving co-planning and co-execution when completing research tasks, suggesting that the two are mutually beneficial while enabling more granular steering of the agent.
4. A 7-day deployment study of COCOA with 7 researchers, where participants made use of self- and AI-assigned steps to flexibly delegate human-AI agency and embed their expertise into the workflow.

6.2 Related Work

6.2.1 Planning and Interactivity in LLM Agents

Chapter 2 surveyed planning as an avenue for configuring and negotiating labor between users and AI agents (Section 2.3.2). Here, we focus on the systems most proximate to our own.

An emerging body of work has started to experiment with more interactive techniques for agents. He et al. [248] separated agent planning from execution and found that user involvement during execution can significantly boost agent performance, but poses a high cognitive load. Kazemitabaar et al. [315] developed interfaces for data analysts to edit an execution plan of an LLM to provide more control points for steering behavior, while Google’s Deep Research [220] can re-generate a plan if the user provides feedback via prompting. However, these interactions are *corrective* rather than *collaborative*—that is, users would typically engage in these interactions to correct agent behavior post-hoc rather than *proactively iterate back-and-forth* with the agent at multiple points in the workflow. We do this by introducing affordances for collaborative human-AI planning and execution via a shared representation of tasks and delegation of agency. While our work shares some conceptual similarities with Kazemitabaar et al. [315], we focus on enabling human-AI co-agency *across steps* in multi-step plans rather than revealing hidden model reasoning *at each step*.

6.2.2 Computational Notebooks

Donald Knuth’s vision of *literate programming* interleaves computer programs with natural language documentation [333]. This vision later gave rise to the computational notebook paradigm, which organizes program imperatives, input data, and rich outputs (e.g., tables, visualizations, interactive widgets) into linearly arranged cells [105, 360]. Computational notebooks are a well-studied paradigm in HCI (e.g., [37, 105, 120, 379, 392, 471, 729]), with most prior work focusing on how data scientists leverage them in their workflows.

In our work, we take inspiration from design decisions made for computational notebooks because of the many parallels between workflows in notebooks and mixed-initiative task completion systems [232, 262, 690]. For instance, data scientists often break down a high-

level data analysis task into executable cells in computational notebooks [316, 561]. As they execute each cell and inspect outputs, they can verify the quality and sensibility of outputs before moving forward in their analysis, or go back to modify existing code cells before execution to interleave between programming, execution, and output examination [317]. Finally, the notebook can serve as documentation and communication of their activities [379].

By drawing these parallels, we can reveal new opportunities to design interactive workflows and interfaces for LLM agents. Just like how a data scientist may easily add new cells in a notebook or reconfigure existing cells to adapt their analysis plan, a user may interactively edit an agent’s plan of execution to better steer the agent in productive directions. Furthermore, an agentic system may simultaneously be more usable and resource-efficient if a user could direct an agent to iterate on a subtask to improve its output before continuing on to subtasks dependent on that output. This also demands new ways of viewing and editing agent outputs on subtasks, which increases opportunities for human input over existing approaches such as simply logging agent actions [36, 494, 547]. Our work exploits these parallels to contribute new design patterns for steering agentic systems. We also apply novel extensions of these parallels for human-agent collaboration: agent-proposed plans (i.e., dynamic AI-generated notebook cells) and agent vs. user steps (i.e., AI executes some cells while the user executes others).

6.2.3 AI for Scientific Research

Significant efforts in recent years have advanced how AI can be used to support scientific research. Of particular interest to us are works that leverage these recent advances to help researchers with planning and execution of *literature-focused tasks*. Planning a research project is a cognitively demanding, complex process consisting of iterative cycles of divergent and convergent thinking grounded in literature review [103, 140, 505], making it an ideal scenario to explore human-agent collaboration. One major thread of prior research has focused on leveraging AI models as tools or components in user-driven systems. These include tools for paper reading [103, 539] and skimming [197], literature review [373, 505], paper recommendation [307, 308], information retrieval and sensemaking [101, 196, 310, 669],

and ideation [39, 234, 287]. On the other end of the spectrum are fully automated end-to-end systems where an AI agent has much more agency in planning and executing literature review tasks [159, 579], or even to attempt carrying out entire research projects on their own [220, 223, 288, 405, 488, 520, 576].

As AI systems become more capable, we ask: Is the generation of full research artifacts (e.g., research questions, proposals) the most desired and promising paradigm to assist researchers? Indeed, when working with other (human) collaborators, researchers often find richness in co-evolving partially completed artifacts [295, 365, 603, 617] and answering or asking questions that stimulate critical thinking and reflection [501, 504]. Further, some fully automatically generated artifacts are still found to be of lower quality than those generated with human involvement [288]. Schmidgall et al. found that even systems designed for end-to-end research automation produce significantly higher quality outputs with human involvement at each stage [576].

In this work, we build on recent advances in AI research support tools, but rather than focusing on generating predetermined research artifacts, we contribute an interaction design pattern that allows for flexible specification of the final artifact and collaboration between a researcher and an AI agent.

6.3 *Formative Study*

Building on prior work, we aim to enhance collaboration between humans and AI agents using a shared, interactive operational representation. We focus on scientific research—a challenging domain for human-agent collaboration given its complex, multi-step workflows and the nuanced decisions researchers make upon processing information-dense scientific papers. Specifically, we investigate the potential for project documents—continuous records of ideas, updates, and tasks—to act as the agent environment. We thus conducted a formative study to answer the following research questions:

F1: What are the properties and opportunities of researchers’ project documents for human-agent collaboration?

F2: How do researchers engage in planning within project documents?

F3: How would researchers prefer to collaborate with an AI agent in their research workflows?

6.3.1 *Participants, Procedure, and Data Analysis*

We recruited 9 Ph.D. students (detailed demographics in Table C.1 of Appendix C.1) through an interest form sent to a research organization’s internal Slack channel and the authors’ professional connections. We targeted Ph.D. students as they often lead the detailed planning and execution of research projects and are the primary editors of project documents.

All studies were conducted virtually over Google Meet and lasted 60 minutes. Before the study, we collected from participants an active or past research project document and a brief description of their research interests. The study was divided into 3 parts:

1. An activity involving the participant’s project document to better understand their current planning behavior (25 minutes).
2. An activity with a Wizard-of-Oz (WoZ) design probe to explore how researchers plan and explore research ideas in a document editor with LLM support via a chatbot (25 minutes).
3. An exit interview where researchers reflected on their experience with the probe and using LLMs in research (10 minutes).

Details for each part can be found in Appendix C.2. Each participant was given a \$35 USD honorarium after the study. The study was reviewed and exempted by our organization’s internal review board. We qualitatively analyzed project documents produced from the study probe activity alongside study transcripts. Our data analysis methods can be found in Appendix C.3.

6.3.2 *Findings*

In this section, we redact any project-specific details for privacy and intellectual confidentiality.

[F1 & F3] Project documents are promising environments for human-agent collaboration

Participants’ project documents acted as a “hub” for their projects. These documents commonly included meeting notes, *planned to-do items*, *progress updates* tied to to-do items, *research questions* to discuss with collaborators, and *links to other documents* detailing a particular project component in greater depth. These documents often served as “scratch paper” for researchers to informally leave a trace of their reasoning and high-level goals, conduct short-term and long-term planning, and track progress of these plans. Most documents (P1, P2–P4, P7–P9) contained one or more problem statements that motivated the project and stated its core contributions. From there, participants listed subgoals and ideas for planned exploration (all participants).

Participants expressed a desire for tighter integration between an AI agent and their project documents in their research workflows for two main reasons. **First**, they wanted to receive in-situ AI support as they plan and execute research tasks in their documents. P1 and P2 both envisioned an AI system “*actively engaging with the content I’m writing [...] after I write each statement, the system could retrieve relevant papers that could provide background or related work for me to read more.*” P9 appreciated “*the ability to lay out the steps when I start something. [It] was very helpful [for] visualizing the outcome.*” **Second**, participants wanted the AI agent to have access to the broader context that already existed in their project documents. When using the probe, P2 was concerned that they would need to “*prompt [the chatbot] with a lot of background,*” but realized they “*might have [the background] written down*” in their document. P3 desired closer document integration but warned that the agent may clutter their document. They suggested: “*having something on the side that does not flow into where I am writing, like a side [panel]—if I want something from it, I want to bring it back into my doc.*”

[F2] Literature search and understanding play a central role in research planning

In all project documents, we saw places where participants intended to initiate a multi-step plan (often via a *how?* and *can we?* question) to address different research tasks. The most

common category of tasks is ones that were literature-augmented (e.g., literature search and understanding), which are often combined with a wide variety of other research actions, such as experiment/artifact design (P3), experiment/code execution (P4), data inspection and synthesis (P7), communicating and discussing results (P5), and ideation (P2). Literature was not only a significant component of *existing* plans, but was important for informing *future* plans. Some participants’ plans (e.g., P3, P4, P7) were formed from hypotheses they hoped to verify, which prior work may have already addressed; P7 pointed out that the hypotheses they had in mind can “*actually be proven from previous papers*” so they saw literature review as a planning aid. Consequently, all participants expressed a desire to use AI to help with *literature-augmented tasks*—exploring and understanding relevant literature to inform decision-making. This finding echoes recent surveys on how researchers leverage LLMs to conduct research, where the most frequent usage category was information-seeking [387].

[F3] Participants preferred to perform higher-level reasoning and synthesis themselves

In addition to highlighting tasks where they wanted AI assistance, participants also spoke about tasks they did not want AI to automate away. In particular, participants saw higher-level reasoning and information synthesis as critical tasks they wanted to do themselves. They were also often unsatisfied with AI’s outputs on these tasks. P3 explains: “*this tinkering process around reading and playing around with things is what gives you the ideas. I don’t know if I want those things automated because the process is as helpful as the final result.*” P5 agreed that they “*wouldn’t want it to be making the final decisions for sure. Just give me inspiration for where I can go.*” Specifically, they shared that “*the main thing I worry about is feeling enough ownership*” if AI makes more consequential decisions. An overeager AI assistant that attempts to perform tasks researchers prefer to do themselves is frustrating because it does not complement the user’s work and instead creates undesired noise for them to filter through.

6.3.3 Design Goals

We synthesize our formative study findings into three design goals for an interactive system that facilitates meaningful collaboration between researchers and AI agents by using plans as a shared representation.

DG1: Provide opportunities for human-agent collaboration across planning and

execution. Agent developers have called for decoupling planning and execution to avoid fruitless execution [281]. Indeed, participants discussed *planning* (i.e., breaking down a problem into smaller, more actionable tasks like they did in their project documents) and execution (i.e., completing steps in the plan and refining outputs, with or without AI assistance) as separate but complementary actions. We thus see planning and execution as two distinct stages for fertile researcher-AI collaboration. Because these two stages are closely intertwined and synergistic, we aim to interleave the two in our system.

DG2: Allow flexible delegation of work between researcher and agent.

Researchers may not want to delegate all parts of a plan to AI [F2, F3]. We also observed that this preference may be context-dependent and constantly shifting. Our system uses a concrete implementation of a theory of flexible delegation proposed by Satyanarayan [572].

DG3: Integrate seamlessly into a document environment.

Project documents contain rich externalizations of researchers’ workflow, thought processes, and plans (e.g., “to-dos”) [F1]. This is key information an AI agent can use to better assist the researcher. The document is also an appealing environment for researcher-agent collaboration, but requires careful information management strategies to prevent unwanted distractions [F3]. Thus, we aim to 1) allow the researcher to use familiar document editing affordances to interact with the agent, and 2) strategically manage outputs to avoid excessively cluttering the document.

6.4 Cocoa: System Walkthrough and Implementation

We present COCOA, a system that embeds an AI agent into a document editor and provides an affordance we call *interactive plans* that allow users to collaboratively plan (**co-plan**) with the agent—the agent proposes an initial plan of execution to tackle a user request that the user can edit to their liking. Then, users can collaboratively execute (**co-execute**) the plan with the agent—the user and the agent can build off each other’s work to synergize human and AI capabilities, and provide flexible delegation of human and AI work.

To illustrate the features and functionality of COCOA, we follow the journey of Nora, a fictional researcher in human-AI interaction, as she uses her project document to further explore open questions in interactive interfaces for AI agents.

6.4.1 Co-Planning

Nora’s project document is an informal, reverse chronological log of research progress and includes information such as meeting notes, rough ideas, links to literature, and questions for herself and her collaborators. She has a specific question in her document that she wants to explore further, but being new to the topic, she is not sure how to proceed on her own: *“What are new ways AI agents can interactively elicit human feedback?”*

Reviewing and selecting an initial plan

To initiate co-planning, she highlights the question to invoke the agent on it using a button that appears in a floating menu. The agent analyzes the question within the context of her entire document and proposes a few different plans via a **plan selector** UI. Noticing one of the proposed plans started with generating a list of more detailed concepts, which would potentially give her the opportunity to learn and tune the specific paper search directions, Nora selects the plan, which then becomes fully interactive within the document (Figure 6.2).

Assigning “Agent steps” and “User steps”

The initial plan included steps for brainstorming relevant topics, searching for papers, and making connections between papers (Figure 6.2). To balance between effort and control,

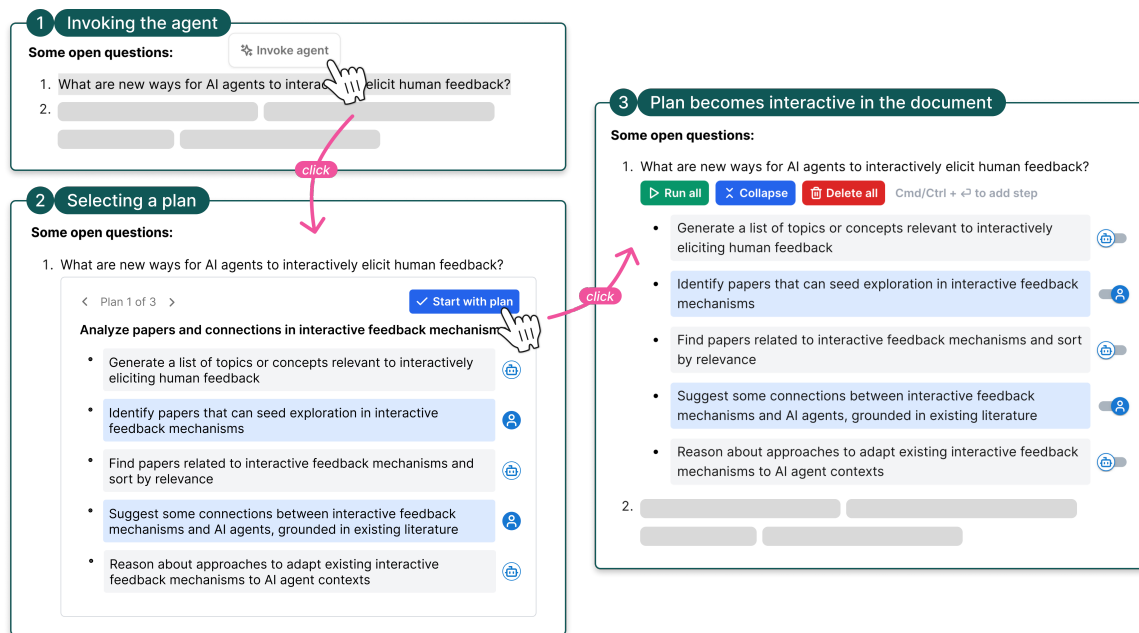


Figure 6.2: A user invokes the agent on a piece of text in the document by clicking on the “Invoke agent” button that appears on hover whenever text is highlighted. The agent will use the highlighted text and context from elsewhere in the document to propose a series of plans, displayed in a plan selector that appears under the highlighted text. Once the user selects a plan, it becomes fully interactive in the document.

Nora uses the **step assignment toggle** (Figure 6.1 B&C, Figure 6.2 right) to assign tasks that are relatively low risk but high effort to the agent, such as searching for papers and expanding on some of her preliminary ideas. In contrast, she assigned tasks that were more consequential or required higher-level thinking to herself, such as identifying seed papers for the agent to explore more broadly or making insightful connections between multiple papers.

Plan Editing and Interactive Replanning

Upon further review of the plan, Nora is now also curious about the prominent authors in the area and wishes to learn about their papers first. She edits the **step description** to reflect this. The step description can be edited just like editing any other bulleted list in her document, making this intuitive for Nora. Similarly, Nora can also easily **add and delete steps from the plan**. For example, she notices that the agent has proposed a summary step that may be irrelevant, so she highlights the step and hits **Backspace** to delete it. Finally, she uses the keyboard shortcut **Cmd/Ctrl+Enter** while selecting the last step to add a new step below it, and edits the step description of the newly added step. These interactions mirror how one would edit items in a bulleted list in text editors, making them intuitive for Nora. While COCOA allows Nora full control over how she wishes to manually edit the plan, this process can be effortful for users. Therefore, the system provides support for *replanning* in the following ways: *First*, when editing an individual step, Nora can request the system to provide suggestions for alternatives, informed by the previous steps and document context. *Second*, after the user has edited some plan step, the plan may become incoherent. For example, a plan step that searches for additional relevant papers given a set of seed papers might no longer make sense if the previous step were edited to produce a list of prominent authors. In this case, COCOA **automatically detects this and replans subsequent steps** accordingly by removing those steps and “autocompleting” the plan with new ones (Figure 6.3).

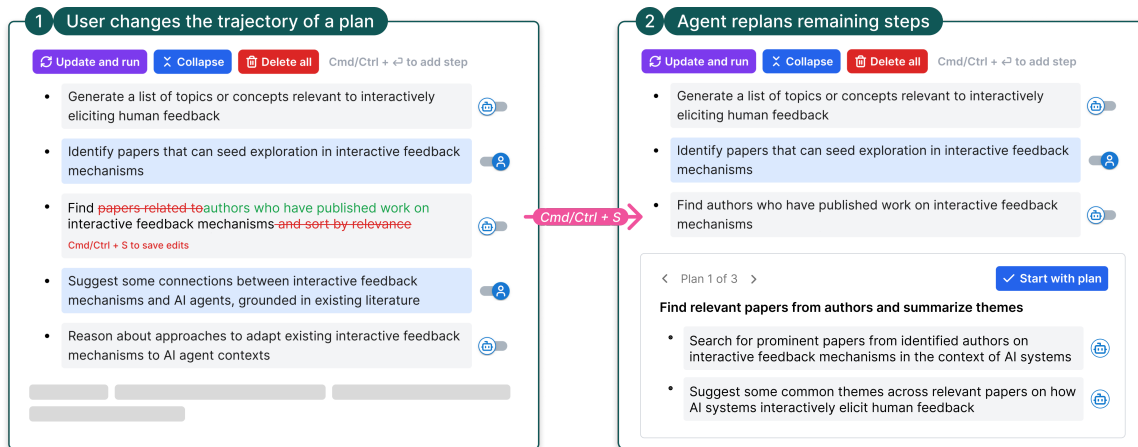


Figure 6.3: If major changes are made to a step that changes the rest of the plan’s trajectory, COCOA detects this and will trigger replanning. Replanning replaces subsequent steps with ones the agent suggested for “autocompleting” the plan.

6.4.2 Co-Execution

Now that Nora has made her desired edits to the plan, she is ready to co-execute it with the agent.

Continuous and stepwise execution

Drawing inspiration from computational notebooks, there are two modes of co-execution: continuous and stepwise. Each step has a **step completion indicator** that indicates whether the step has not yet been run (a bullet point), is in progress (a spinner), requires user input (a question mark badge), or is complete (a checkmark). This notebook-inspired design allows Nora to dispatch the agent to run autonomously on parts of a plan she trusts the agent to complete on its own. She can also iterate on specific parts of a plan by examining and potentially amending the output of each plan step before continuing.

Nora wishes to run through the entire plan once to see the output. She triggers *continuous co-execution* with the **[Run all]** button at the top of the plan. In **continuous co-execution**, the agent will automatically continue onto the next step as soon as it (or the

user) has completed the previous one, until it reaches a user step or the end of the plan. By contrast, if she wishes to iterate and refine the outputs of individual steps before the agent moves onto the next one, she can opt for **stepwise co-execution**.

Nora can flexibly switch from stepwise to continuous co-execution at any point by clicking the **[Run remaining]** button at the top of the plan. Conversely, Nora can switch from continuous to stepwise co-execution by clicking on the **[Pause after this step]** button during execution and manually running subsequent steps.¹

Editing outputs of agent steps

As the agent completes a paper search step, Nora clicks into that step to examine its output. This opens a sidebar to the right that is populated with interactive paper cards (Figure 6.4). Nora can further guide the agent with her expertise by removing papers she considers irrelevant and/or adding papers the agent missed using a built-in paper search functionality connected to the Semantic Scholar academic database [328]. For now, she removes a couple of papers in the output that were not published recently and were published in a less relevant venue.

The interactive UI in the sidebar dynamically adapts according to the step’s *output type*, which falls into one of five categories in COCOA: **papers**, **authors**, **topics**, **entities**, and **text**. Papers, authors, and topics are drawn from Semantic Scholar and are displayed as interactive cards that Nora can add, delete, or search for using the sidebar’s built-in Semantic Scholar search. Entities are lists of items in natural language (search queries, research questions, etc.) and are displayed as editable pills. Text is natural language text displayed in an editable text box.

If the agent fails to complete its step (e.g., it finds no papers or runs into an error), COCOA will also proactively alert Nora so that she can complete the step instead, or retry with the agent.

¹COCOA does not permit out-of-order stepwise execution because some steps may rely on the outputs of previous ones. We also note that out-of-order execution in computational notebooks is a major user pain point identified in prior academic work [105, 360] and industry reports [233, 299].

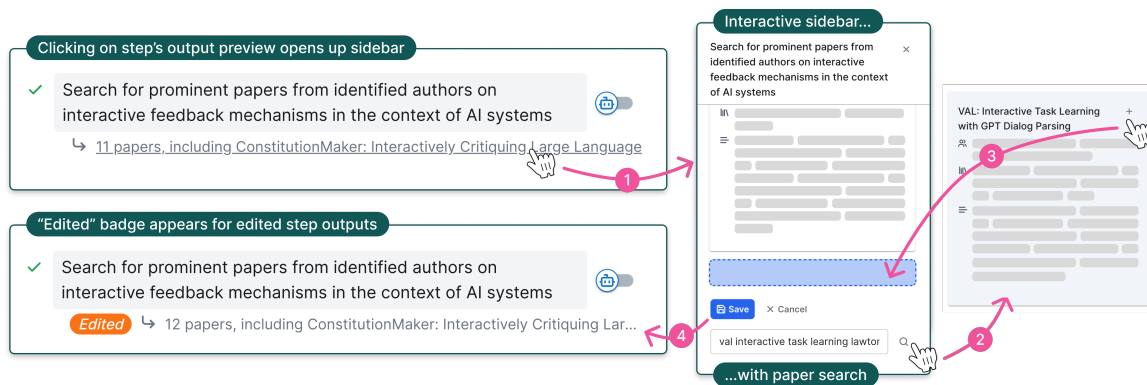


Figure 6.4: Users can click the output preview text beneath the plan step to access the interactive sidebar (1). By editing agent outputs in the interactive sidebar (2), users draw from their expertise to edit outputs, such as adding papers the agent might have missed (3), to guide the agent. Once edited, an indicator will appear in the output’s text preview below the step (4).

Completing user steps

Before executing the plan, Nora had assigned step 2 to herself during co-planning. She wanted the agent to start a broad search of many relevant topics and concepts, but then wanted to contribute some seed papers she knew would be relevant to steer the agent’s subsequent actions.

COCOA has now paused at step 2, highlighting the step in orange to attract Nora’s attention and to seek her input. When she clicks into that step, the same interactive sidebar opens to the right (Figure 6.4). She could then manually review the topics generated in step 1 and use the paper search feature built into the sidebar to browse relevant papers and add them to the output of step 2. Additionally, she recalls a few papers that a collaborator had recommended to her previously, finds them using the paper search feature, and adds them to the output.

Final output panel

After the final plan step has been executed, the agent adds a **plan output panel** (Figure 6.5) containing a modified version of the last step’s output into the document just below the plan. The modification involves rewriting the output, given the context of the plan and its outputs, to more directly connect to the original user request. This brings the results of running the plan into the document so Nora can easily reference it in the context of other content. The panel is collapsed by default to save space and reduce clutter, but she can expand it to copy and paste parts of the output she finds useful and add them to her document. If she does not want the panel in her document at all, she can simply delete it; she can still access the outputs of all steps in the sidebar.

6.4.3 Interleaving Co-Planning and Co-Execution

In addition to supporting co-planning and co-execution independently, COCOA is designed to support **synergistic interleaving of the two**. As Nora co-executes the plan with the agent, she encounters a relevant author that the agent identified and wishes to further explore the author’s papers. She edits the following step’s plan description to satisfy this. She reruns that step and receives a new batch of papers, and the agent automatically reruns the rest of the plan with the updated output. She is also not satisfied with the agent’s interpretation of common themes and believes that she can surface deeper insights by reading the papers herself, so she toggles that step to be a user step. After reading the papers in detail, she returns to her document and jots down her insights. She now finds the next step repetitive, so she deletes it. Overall, her use of COCOA is highly iterative, and she smoothly transitions between co-planning and co-execution.

When designing COCOA, we were inspired by how the computational notebook paradigm supported scientists via tight coupling of code and output, enabling rapid iteration and steering of complex scripts. To summarize this connection, we map COCOA’s features to those in computational notebooks in Table 6.1.

1. What are new ways for AI agents to interactively elicit human feedback?

▶ Run all
✕ Collapse
🗑 Delete all
Cmd/Ctrl + ↵ to add step

- ✓ Generate a list of topics or concepts relevant to interactively eliciting human feedback
 - ↳ 10 topics, including Direct Manipulation Interfaces and Adaptive Control (AC)
- ✓ Identify papers that can seed exploration in interactive feedback mechanisms
 - ↳ 5 papers, including Principles of mixed-initiative user interfaces and Guidelin...
- ✓ Find authors who have published work on interactive feedback mechanisms
 - ↳ 20 authors, including E. Chi and Besmira Nushi
- ✓ Search for prominent papers from identified authors on interactive feedback mechanisms in the context of AI systems
 - Edited ↳ 12 papers, including ConstitutionMaker: Interactively Critiquing Lar...
- ✓ Suggest some common themes across relevant papers on how AI systems interactively solicit human feedback
 - ↳ Text with 261 words: AI systems that effectively solicit human feedback shar...

📄 **Plan Output** 📄 🗑

AI systems that effectively solicit human feedback share several common themes, as highlighted in recent research papers.

One key theme is the **structured task breakdown**, which helps reduce th...

▼ Show more

Figure 6.5: Once execution of the entire plan is complete, the agent inserts a plan output panel in the document with the last step's output. This allows the user to view the plan's results within the context of their document. The panel can be expanded or collapsed, and remains in the document even if the plan itself is collapsed for easy access.

COCOA Feature	Computational Notebook Feature
Editable plan steps that can be composed into a larger, more complex task.	Code cells that can be composed into a larger, more complex program.
Users can choose between stepwise and continuous execution.	Users can choose between running individual cells or ‘run all’.
Users can view and improve step outputs within the plan.	Users can view code outputs within the notebook.
Users can flexibly switch between improving the plan and executing the plan steps.	Users can flexibly switch between improving the code and executing code cells.
Plan steps assigned as user steps pause execution so users can perform tasks themselves.	Breakpoints pause execution so users can examine current code outputs; all cells are executed programmatically.
Completed plan can serve as documentation of agent & human work.	Completed notebook can serve as documentation of analyst’s work.

Table 6.1: Main connections between features in COCOA and those in computational notebooks.

6.4.4 Implementation Details

COCOA is implemented as a web application with a Next.js and TypeScript frontend communicating with a Flask backend. The frontend uses the Tiptap² framework for the main document editor. Each document supports synchronous collaboration via Hocuspocus³ and all changes are auto-saved to Tiptap Cloud, which also serves as storage for all participant documents. Interactive components within the document editor are implemented as custom Tiptap extensions in TypeScript.

The Flask backend orchestrates LLM activity with calls to a custom tool-calling LLM

²<https://tiptap.dev/>

³<https://tiptap.dev/docs/hocuspocus/>

agent⁴ powered by GPT-4o and scaffolded by LangChain.⁵ Our agent is designed to assist researchers through literature-grounded response generation. To do this, we implemented tools that give our agent access to millions of research papers via publicly available Semantic Scholar APIs.⁶ This API is powered by The Semantic Scholar Open Data Platform [328] which consists of a state-of-the-art PDF extraction and knowledge graph normalization pipeline. This gave our agent access to both the full text of millions of research papers and also their rich metadata, such as authors, fields of study, and publication venues [328]. More implementation details can be found in Appendix C.4.

6.5 Lab Study

We conducted a within-subjects task-based lab study with 16 researchers to evaluate COCOA against a custom chat baseline. Specifically, our study aimed to address the following research questions:

- L1:** How does COCOA compare to our chat baseline for ease of use, steerability, and general utility⁷ in research project documents?
- L2:** How does interleaved co-planning and co-execution contribute to ease of use, steerability, and general utility in COCOA, if there is a significant improvement over the baseline?
- L3:** What kinds of steps did researchers wish to assign to an agent and themselves in practice?

6.5.1 Participants

We recruited 16 Ph.D. and postdoctoral researchers (14 Ph.D.s, 2 postdocs; 10 female, 6 male) in computer science (CS) or CS-adjacent areas via university mailing lists, word of

⁴<https://platform.openai.com/docs/guides/function-calling>

⁵<https://www.langchain.com/>

⁶<https://www.semanticscholar.org/product/api>

⁷General utility is a composite metric created from Q1, Q2, and Q4 from Appendix C.8.

mouth, social media recruitment messages (on Twitter/X, Mastodon, Bluesky), and personal connections. We recruited on a first-come, first-served basis as we conducted our studies and closed recruitment when we approached data saturation. Further details about our recruitment process, pilot studies, and participants can be found in Appendix C.6 and Table C.2 of Appendix C.5.

6.5.2 *Baseline System*

As our primary research question is to better understand the costs and benefits of COCOA’s co-planning and co-execution affordances, the baseline system (Figure 6.6) was a common chat interface powered by the same underlying LLM agent: a custom tool-calling agent powered by GPT-4o with access to the same state-of-the-art tools for searching, analyzing, summarizing, and comparing literature across millions of papers on Semantic Scholar [328], in addition to standard LLM capabilities. See C.4 for more details about the agent. The design of the chat interface closely resembled that of existing AI-powered research systems (e.g., Assist mode in Scite [579], Google Scholar Labs, various deep research systems), and popular LLM chatbots (e.g., ChatGPT, Claude). During the study, participants positioned the chat interface beside the baseline document editor, which is the same editor that COCOA uses, except we removed the option to invoke the agent to eliminate all co-planning and co-execution interactions.

We chose this baseline because it mimicked the dominant interaction paradigm implemented in AI-powered research tools at the time of our study and allowed for strong comparisons on aspects like ease of use. We recognize that variations of this baseline, such as forcing the agent to show an editable plan before execution (e.g., [248]), are also strong baselines that can help further isolate the factors contributing to our results. We discuss ablations of COCOA as alternative baselines in Section 3.9.

6.5.3 *Research Task and Study Procedure*

To make the study more realistic and grounded in real-world research projects, tasks for this study were open questions or unfinished items from participants’ existing project documents.

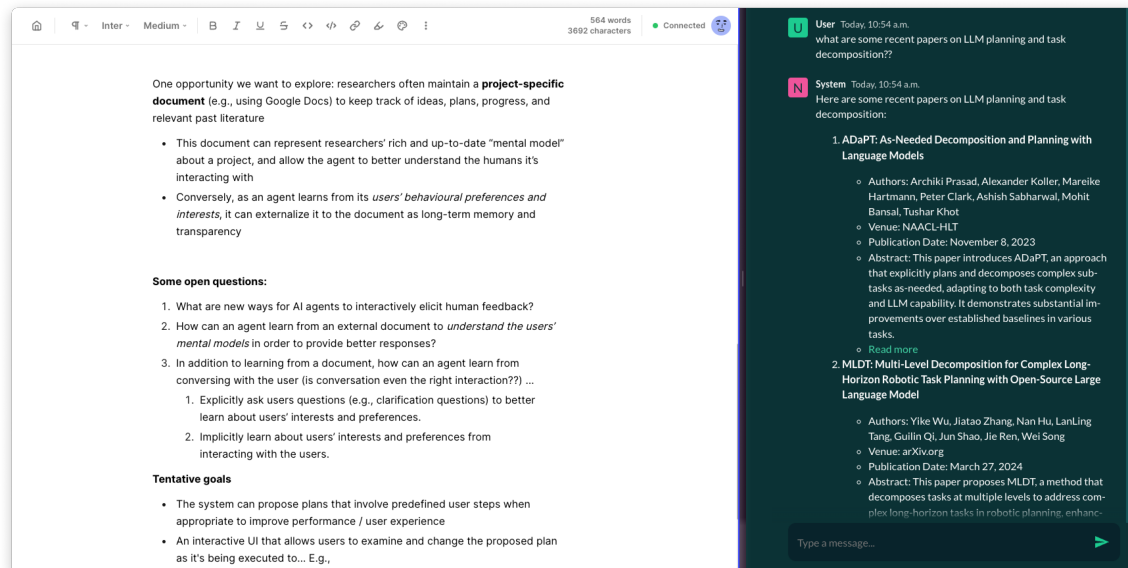


Figure 6.6: Participants' typical setup for our baseline condition. On the left, the same document editor from COCOA but without an option to invoke the agent for co-planning and co-execution. On the right, the agent is situated within a chat interface.

Details for how we controlled for the length of the study and ensured the two tasks were valid and comparable can be found in Appendix C.7. Tasks were randomly assigned to COCOA and the baseline.

The first author (study facilitator) conducted 1:1 studies with the 16 participants over Google Meet between October and December 2024. Each study was 90 minutes in length. The study started off with introductions and a brief overview, followed by the two tasks, which were counterbalanced across participants. Detailed procedures for each condition can be found in Appendix C.7. After each condition, participants filled out a short evaluation form with 5-point Likert scale questions for that condition (see Appendix C.8). These questions were inspired by the System Usability Scale [76] but adapted for the research context.

The task-based portion of the study was 75 minutes. The study concluded with a semi-structured interview that lasted around 15 minutes. Participants were asked to reflect on their experiences across the two systems and discuss the pros and cons of both. They were also asked about particular decisions the study facilitator observed when using each system.

Participants received a \$75 USD honorarium upon completion of the study. All studies were recorded and transcribed by Google Meet. This study was reviewed and approved by the Allen Institute for AI’s IRB.

6.5.4 Data Analysis

The first author analyzed the recording transcripts using Braun and Clarke’s reflexive thematic analysis [74]. This approach uses a hybrid inductive-deductive approach to iteratively surface codes and themes across the data. We paid close attention to codes related to participants’ comparisons of their experiences across the two systems and gradually grouped them into themes. NotebookLM⁸ was used to help iterate on themes and discover new ones. We performed statistical tests of participants’ ratings on the evaluation forms with the Wilcoxon signed-rank test (with the Bonferroni correction for our multi-rating analysis on significant results) given the non-parametric nature of our data. We saved all documents used in the

⁸<https://notebooklm.google.com>

study as well as all conversation histories in the baseline system, and referred back to them when needed.

The first author also coded all video recordings from the study to note the timestamps at which participants engaged in and disengaged from a particular interaction. In COCOA, these interactions were **co-planning** (see 6.4.1), **co-execution** (see 6.4.2), and **output inspection** (when the participant passively inspected the system’s outputs). The closest equivalent interactions were used as codes in the baseline: **prompting for planning** (instructing the agent to perform an action without any attempts to modify the agent’s earlier outputs), **prompting for editing**⁹ (instructing the agent to modify or only consider a subset of its output in future actions), and **output inspection**.

We established these equivalences based on the functional purpose of the interactions: co-planning and prompting for planning both instruct the agent to produce new outputs, while co-execution and prompting for editing aim to improve an existing output. We note that there is no perfect correspondence between interactions across the two systems. For example, co-execution is not only about refining agent outputs, but also giving users agency to complete tasks they assigned for themselves. We acknowledge this by shading the interactions in COCOA and the baseline with slightly different colors in Fig. 6.7. However, because co-planning and co-execution subsume the functional roles of their baseline counterparts in the context of our study, we consider them appropriate to compare.

6.6 Lab Study Results

A high-level overview of participants’ interactions with COCOA and the baseline is depicted in Figure 6.7. In both conditions, output inspection occupied the most time out of the coded interactions, but was lengthier in the baseline than in COCOA (54.1% vs. 33.6% of the approximately 25-minute session). A paired t-test found this difference to be significant ($t(15) = 3.95, p < 0.001$). Participants also engaged in co-execution in COCOA significantly more often than the closest equivalent interaction—output editing via prompting—in the baseline (15.2% vs. 3.1%, $t(15) = 4.29, p < 0.001$). They spent slightly more time on co-

⁹Since there were no user steps in the baseline, the closest equivalent interaction is attempting to edit the agent’s output via prompting.

planning (21.5% vs. 17.4%), although this was not significant ($t(15) = 1.01, p = 0.33$). Our analysis indicates that, when provided with more affordances to interactively modify the agent’s outputs, users will take advantage of those opportunities for active engagement and shift away from passive output consumption. That is, users could **interleave co-planning and co-execution to steer the agent through tighter feedback loops**, rather than relying on slower cycles of trial-and-error prompting and output inspection.

We also computed performance metrics for both co-planning and co-execution from coded interactions and system logs. These included the number of step edits, reassignments, and output edits across all participants. Table 6.2 displays our results. Overall, 32.6% of steps were edited and 17.4% were reassigned. This shows that most plan steps proposed by the agent (the content of the step as well as the assignment) were accepted as-is. During execution, the agent surfaced an average of 12.9 discrete output items (i.e., papers, authors, topics, text entities) per step to users. On average, 2.1 (16.3%) were deleted by participants, suggesting that the system had reasonable precision.

However, these metrics alone may not paint a holistic picture about COCOA’s use. To more deeply understand the context and rationale behind users’ actions, we analyzed videos from our user studies (Fig. 6.7) alongside qualitative transcripts from participants thinking aloud (Section 6.6.2).

In the rest of this section, we present quantitative and qualitative analyses of participants’ data from our user study. We group these results by our research questions from the start of Section 6.5. We denote the median rating of our baseline and COCOA as M_b and M_c , respectively, and the Wilcoxon test statistic as W .

6.6.1 Steerability, Ease of Use, and Utility (L1)

Introducing novel interactive systems with additional affordances can often lead to higher effort when using the systems as a trade-off for better utility [278,602]. However, based on participants’ post-task ratings on a 5-point Likert scale, there was no significant difference in perceived *ease of use* between the baseline and COCOA with 16 participants ($M_b = 4.5, M_c = 4, p = 1.000, W = 7.50$). At the same time, we observed a significant difference that COCOA

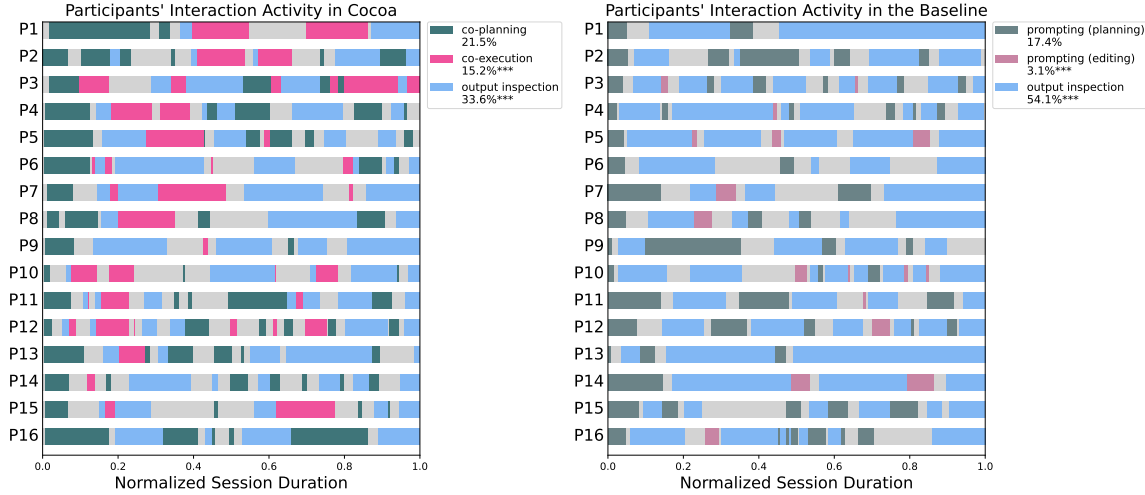


Figure 6.7: Participants' interaction activity in COCOA and the baseline within the study sessions. *** indicates a statistically significant difference ($p < 0.001$ via a paired t-test) with the closest equivalent interaction in the opposing condition. Gray areas represent spans of time where the participant was not engaging in any of our coded interactions because they were answering a question posed by the study facilitator and/or waiting on the system. Definitions for all coded interactions can be found in Section 6.5.4.

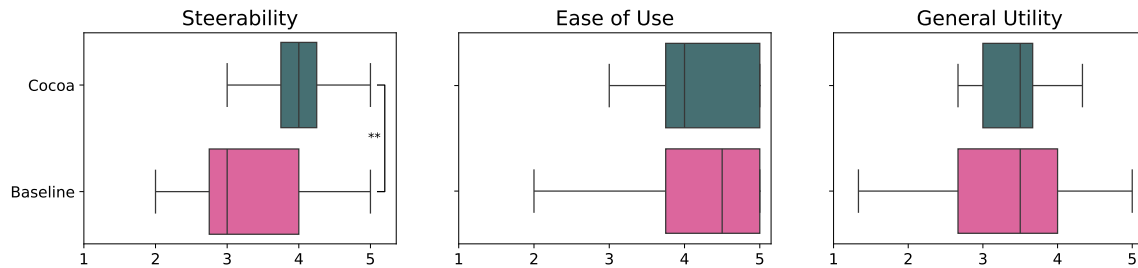


Figure 6.8: Participants' Likert scale ratings of steerability, ease of use, and general utility across our baseline and COCOA. ** indicates $p < 0.01$ via a Wilcoxon signed-rank test.

Metric	Count (% , if applicable)
Total steps completed across all plans	86
Step edits by user	28 (32.6%)
Step reassignments by user	15 (17.4%)
Step additions by user	12 (14.0%)
Step deletions by user	4 (4.7%)
Replanning triggered by system	14
Avg. output items (i.e., papers, authors, topics, text entities) per step	12.9
Avg. output items deleted per step	2.1 (16.3%)

Table 6.2: COCOA usage metrics from our lab study.

provided better *steerability*. In response to the question “*I could easily steer the system towards doing something helpful,*” participants rated COCOA higher than the baseline to a significant degree ($M_b = 3, M_c = 4, p = 0.005, W = 0$),¹⁰ with $W = 0$ indicating that all participants rated COCOA’s steerability as greater than or equal to that of our baseline. In sum, these results suggest that **COCOA affords interleaving co-planning and co-execution via rich interactive affordances without sacrificing ease of use**, when compared to the simple chat interface used in the baseline condition (Figure 6.8).

We also tested for differences in *general utility*, which was a composite metric consisting of three measures broadly related to the usefulness and insightfulness of system outputs, but we found no significant differences ($M_b = M_c = 3.5, p = 0.7, W = 40$). This was unsurprising—the systems’ utility to the researcher may depend on a range of system-agnostic factors beyond our control (e.g., contextual interpretations of outputs by researchers, the extent to which researchers thought about the task before the study, etc.). However, improved steer-

¹⁰Here, we set the significance threshold to $\alpha = 0.05/3 = 0.015$ using Bonferroni correction. Our result is significant post-correction: $p = 0.005 < 0.015$.

ability may have allowed users to steer the agent away from providing downright unhelpful outputs in COCOA, as indicated by fewer lowly-rated outliers (see Figure 6.8).

6.6.2 Interleaving Co-Planning and Co-Execution Created an Effective Steering Loop (L2)

Outputs from co-execution drove re-planning

Qualitative results based on think alouds and post-interviews suggest that COCOA was valuable not only because it enabled human-agent co-planning or co-execution, but because *interleaving the two meant that co-execution outputs could be used to inform collaborative re-planning*. Rather than treating the plan as a static artifact, participants used co-execution to find interesting and potentially productive directions to steer the agent towards. For example, P7 and P15 both used outputs from agent steps as a signal to “*organize thoughts and iterate on ideas*” (P7). P15 specifically iterated on an intermediate “paper search” step by re-executing it to cover papers from venues that they preferred. This led to subsequent plan updates that ultimately resulted in a more relevant final output. Similarly, P6 observed that “backtracking” to a specific step and editing it was more helpful than digging through a chat conversation: “*With chat, when it does something wrong there’s not really an easy way to fix it because there’s no concrete steps that it’s following, whereas [with COCOA], I can go back and be like, let me have it do something else here.*”

Besides modifying intermediate steps, participants also added steps at the end of the plan to assign follow-up investigations, or deleted steps to sharpen the plan’s focus. For example, P12 added a step at the end of their plan to have the agent suggest some potential next steps to pursue after seeing its paper summaries, while P5 removed a less relevant intermediate step on searching for papers related to autonomous vehicles to avoid distractions from this tangential topic. In general, iterative output evaluation and re-planning allowed participants to better “fine-tune” the agent’s outputs and also their own research process—P1 likened their experience to using “[*plan steps*] as building blocks” that they could mix and match to create a custom workflow.

Fluid movement between co-planning and co-execution enabled granular steering

The interleaving of co-planning and co-execution allowed participants to move fluidly between the two and exert control exactly when necessary. By opting for stepwise execution of the plan, participants could pause the agent to “catch up” with agent activity and ensure outputs are up to standard before continuing. Indeed, P4 and P11 both *verified* and *manually curated* the list of papers returned by the agent from a paper search step before proceeding. P11 specifically cites resource efficiency as a practical reason: *“If I update [outputs of] the first or second step [after executing all steps], it would need to rerun the following steps. I just want to reduce some API usage.”* P10 similarly mentioned that they would “waste energy” running subsequent steps if they later detected errors in initial steps. Overall, many participants chose stepwise execution because it *allowed more room to interleave co-planning*; this in turn offered more granular control over the human-agent collaborative work trajectory and prevented error propagation.

There were, however, still reasons for preferring continuous execution. P8 and P9 thought continuous execution was more efficient because it parallelized human and AI efforts: *“Running all [steps] is a little bit of parallelization in my head where I can get the next thing rolling, and if I end up not having to edit it, that means we’re already moving ahead”* (P8). In this case, P8 does not feel a need to be “caught up” with the agent’s work because they know they can still steer the agent at their own pace by modifying the agent’s execution and triggering re-planning at any point.

Interleaving co-planning and co-execution via direct manipulation in particular also contributed to granular steering. Many participants, including P6, liked the precision with which they could edit the output of a particular step compared to a chat interface: *“With a conversation agent, even if I’m really good at prompting it, I have to redo the prompt and keep changing that original prompt.”* (P6). This feature also allowed P2 to quickly re-plan and repurpose the output when the agent produced unexpected (albeit helpful) paper search queries: *“I was looking for more application intervention based stuff, but this is great because now I can actually use these keywords to broaden my horizon of how I was thinking about these solutions, which is I think interesting. I’m just trying to remove a couple [for future*

steps/.” Finally, participants found that interleaved co-planning and co-execution scoped tasks in a way that naturally afforded more precise control, especially when paired with direct manipulation, as P15 shared: “[COCOA] was easier to control because [the outputs] are so specific. My control is scoped down to these very small tasks. If I want to add more papers, that’s easy to do.”

6.7 Deployment Study and Results

In addition to interleaving *co-execution and co-planning*, we were also interested in studying *the delegation of human and AI work*. Thus, we tracked which steps participants assigned to the agent vs. themselves.

In our formative study (Section 6.3.2), participants preferred to assign tasks related to information retrieval and brainstorming to the agent while keeping higher-level reasoning and synthesis tasks for themselves. In our lab study, however, most participants toggled all plan steps—even higher-level ones assigned to them by default—to the agent. Some avoided user steps because they were curious about “*how the bot does it*” (P1). Understandably, testing the agent’s capabilities and limitations is a natural tendency when learning to use a new system. Many speculated that they would leverage user steps if they had more time to interact with the system outside of the lab environment. P8 explicitly said that “*I assign everything to [the agent] because I don’t think we have that much time for me to refine my [own steps]*.”

To obtain a more holistic understanding of how COCOA can facilitate flexible delegation of human-AI work and more generally assist researchers day-to-day, we conducted a 7-day field deployment study with 7 participants.

6.7.1 Participants and Procedure

We invited all participants from our lab study to participate in this field deployment. 7 participants (P3, P5, P10, P11, P12, P15, P16) agreed to participate.

Participants used COCOA to work on a real-world, in-progress research project of their choice over a 7-day period. Before the study, the study facilitator sent each participant

a user manual¹¹ documenting COCOA’s functionalities. The manual also contained some general guidelines for the week (see Appendix C.9), including aiming to spend at least 90 minutes with the system throughout the duration of the study. Each participant could log onto their personal workspace in COCOA with a set of credentials provided by the study facilitator. We used system logs to verify that there was at least semi-regular activity for each participant throughout the study.

At the end of the 7-day period, participants completed a 30-minute semi-structured exit interview where they were asked about their general experience with the system as well as specific co-planning and co-execution interactions the study facilitator identified from within their workspace. All interviews were recorded, transcribed, and analyzed with the same qualitative analysis procedure described in Section 6.5.4. Participants received a \$100 USD honorarium upon completing the exit interview.

6.7.2 Field Deployment Study Results

COCOA was most useful for literature synthesis and early-stage project planning

All participants agreed that COCOA was especially helpful for **literature discovery and synthesis**. This illustrates the value of interleaving human-AI efforts across planning and execution in complex information foraging and sensemaking activities. P15 found the structured plan for conducting literature search to be *“really helpful for refining my thought process because it made me think more carefully about what I’m actually trying to look for.”* P10 agreed, and also felt that the combination of plan steps involving search query generation and paper search left them with *“more faith that I was getting a good coverage of the field.”* P11 also appreciated COCOA’s speed at literature search and synthesis and found it **conducive to early-stage explorations**: *“if I were to come up with a summary of 20 papers, it takes probably a day, but [in COCOA] it’s around a minute which is really useful to formulate early-stage research ideas.”* Beyond literature search, the structure of the plan was also useful to *“break down tasks and provide a sense of progress”* (P16) and *“identifying important actionable steps I could take with a research problem”* (P12). Overall, participants

¹¹See Supplementary Materials.

found COCOA valuable in their work, with P3 in particular sharing that *“I have an entire tab in my Google doc where we have stuff from COCOA.”*

Step assignment strategy changed: more delegation of work with longitudinal use

Compared to the lab study, where steps were mostly assigned to the agent, participants in the field study assigned more plan steps to themselves. This was likely due to a combination of increased familiarity with the agent’s capabilities and system features, as well as more opportunity to reason about which steps they wanted to keep for themselves and why. These user steps often involved high-level strategic decision-making, writing and developing arguments, running experiments and evaluations, deep reading of literature, and articulating core research ideas and novelty.

The ability to assign steps to the agent and themselves helped participants **create a back-and-forth workflow with the agent**. For example, P15 first figured out *“broad buckets of literature or themes or topics that I need to talk about”* before passing those to the agent to *“help me brainstorm ways to connect those bodies of literature.”* P3, who had already conducted some interviews for their project, first had the agent look up key literature before assigning themselves a step to *“pass in more of the dynamics from my interview notes”* and then had the agent propose interventions based on the literature and notes. For some participants, user steps came later in the plan because of their high-level nature. P11 often assigned the last plan step to themselves because *“it was about the main innovation in the project idea [...] no one has [done it] before so I don’t trust the agent.”*

Participants also indicated that their step assignment **depended on the stage of research they were in**. P3 admitted that *“if I was just doing initial discovery and study planning, I would have gone more agent [steps].”* These findings suggest that participants benefited from the ability to flexibly configure agency in their human-AI workflows.

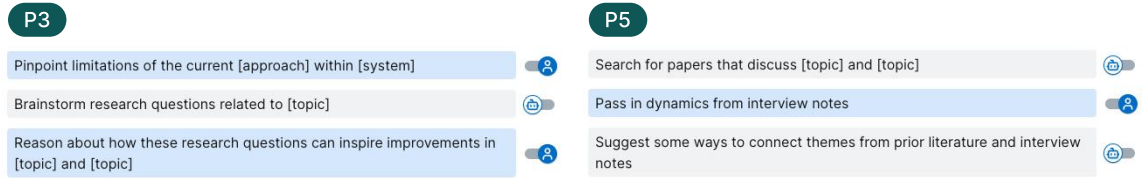


Figure 6.9: Examples from our field deployment where user and agent steps were interleaved in participants’ plans. The plan steps were rewritten to redact project-specific details.

Interleaving co-planning and co-execution supported evolving research trajectories

Just like in the lab study, participants valued interleaving of co-planning and co-execution, but the value for supporting **flexible and iterative workflows** became more apparent with longitudinal use. As participants saw the results of execution, their understanding of the problem space shifted, which led them to want to **anchor certain steps and outputs as “checkpoints,”** and use those as a **springboard for exploring new territory**. For example, P12 shared that *“I don’t need to run the [earlier] retrieval step again. I know that those papers are relevant”* but instead could focus on *“only improving the second step [...] to identify something different from the literature.”* Similarly, P16 felt like they could easily “debug” suboptimal parts of the plan: *“by forcing these checkpoints of brainstorming search queries and assembling lists of papers, it helps me get a sense of what parts of the process are good and which parts need work.”* Outputs from co-execution also inspired new approaches for participants to **“iteratively build upon a plan or reshape a plan”** (P5). Participants even mentioned that they wanted similar interleaving strategies integrated into other LLM-powered systems; for P11, it was *“one of the improvements I want to see in LLM [reasoning] models”* so that users can force an early stop to the reasoning or steer it in more productive directions. Overall, our deployment of COCOA suggests that interleaving co-planning and co-execution helps support the non-linear nature of scientific inquiry.

Additional desiderata

Participants identified several desiderata for COCOA after spending more time with it. Some participants wanted to clearly see **what context COCOA used** to generate plans and execute plan steps. For example, P10 was unsure whether the system was “*using the context from my other plans to generate the [other] plan.*” The system indeed was, but did not visualize relevant context in the document. P3, P15, and P16 all wanted more transparency into the shared context pool (Section 6.4.2) used across all steps and the specific context drawn upon by each step. Several participants desired **tool interoperability**, such that they could swap out COCOA’s default tools for their own ones (e.g., an LLM finetuned on paper summaries instead of COCOA’s paper summarizer). This way, COCOA would serve as a “hub” for their custom tools. Finally, P5 took inspiration from Notion’s ability to nest pages to suggest nested plans, where plans can be organized and expanded in a hierarchical manner. These suggestions all provide valuable directions for future work.

6.8 Discussion

6.8.1 Design Considerations for Co-Agency in Human-Agent Collaboration

Researchers have called for reframing the design of human-AI systems as not merely the specification of affordances, but construction of open design spaces that support human-AI *co-agency* [572]. That is, rather than providing a fixed configuration of human and AI agency across all interactions, users can flexibly vary the agency they wish to assert and have the AI modulate its agency in response. COCOA offers an early instantiation of this vision. We now reflect upon several key design decisions and discuss how future systems may build upon them or explore alternatives when designing for co-agency in human-agent systems.

We expect that *fluid re-planning* will continue to be important and challenging to design for in future systems. Plans are bound to shift as the user reviews agent outputs and becomes inspired by new ideas. For example, researchers in both our lab and deployment studies dynamically re-planned with the agent upon gaining more clarity on *what they actually wanted to explore* after co-executing earlier steps (see Sections 6.6.2 and 6.7.2). In

COCOA, re-planning *replaces* outdated steps with new ones. However, participants sometimes expressed a desire to preserve the old steps because their outputs can still be useful to reference in the future. We thus see *plan branching* as a promising design pattern for exploring counterfactual plan paths. For example, systems can allow users to swap certain steps in and out of the plan, or set checkpoints at specific steps to anchor exploration. By doing so, useful work can be better retained while supporting flexible plan redirection.

Relatedly, the *length of the plan should be calibrated depending on the task*. Some tasks, such as open-ended brainstorming, benefit from shorter, partial plans where next steps can be flexibly determined based on prior execution. Indeed, a few participants expressed a preference for a step-by-step planning approach over COCOA’s default plans when conducting brainstorming, where they did not know what the next step would be until they saw the results of the previous step. On the other hand, for tasks with more structured and predictable workflows, such as literature review, participants appreciated interacting with a full plan. We calibrated the lengths of plans generated by COCOA based on participants’ plans in our formative study. We recommend that designers of human-agent systems consider what the appropriate plan length is for tasks they hope to support, and calibrate accordingly.

Finally, *user expertise and upskilling potential should be considered* when designing for an appropriate level of agent control. As surfaced in our formative study (Section 6.3.2), there are two primary reasons why users prefer keeping a task for themselves over delegating it to the agent. *First*, the user can perform the task more effectively than the agent. This is often the case where the user holds more expertise (e.g., brainstorming research questions in a user’s primary research area). *Second*, the task is intrinsically valuable for the user to do themselves, even if an agent is capable of automating it (e.g., reading seminal papers to develop research taste). If a human-agent system is designed for expert users who want to regularly hone their skills and intuition, providing many levels of control throughout agent planning and execution (as in COCOA) may be appropriate. If the system is for navigating unfamiliar domains with agents and users are not looking to upskill, fewer controls may enhance user experience. Because interactions such as co-planning and co-execution can burden the user with additional cognitive load, determining the user’s capacity and willingness to take on more cognitive load can inform the appropriate level of control to

afford.

These design considerations were drawn from the specific definition of agency (from Bennett et al. [56]) and domain (scientific research) specific to our study. The design space is vast, and we are excited to witness the application, refinement, and even refutation of these considerations as future work navigates this space with alternative interpretations of agency in different domains.

6.8.2 The Agent Notebook Paradigm?

The design of COCOA’s interface was partly inspired by the computational notebook paradigm. Just like how notebooks help break apart a script into smaller components for unit testing and quick iteration, our “agent notebook” design—featuring plan step “cells” and options for stepwise or continuous execution—helps bring similar benefits to agentic workflows. We saw in both our lab study (Section 6.5) and field deployment (Section 6.7) that participants valued the ability to execute steps one at a time and interactively refine outputs before moving on to subsequent steps. We now discuss agent notebooks as a *potentially generalizable design pattern* and consider its applications to agentic systems beyond COCOA.

Consider “deep research” systems that assemble a detailed report on a particular topic upon searching for and synthesizing information from around the web [220, 488, 520]. These systems may clarify user requests in a chat conversation [488], or allow the user to prompt the agent to regenerate the plan [220] before starting the research process. However, there is no room for user involvement thereafter. Agent notebooks can be particularly valuable here because they afford iteration and control over *specific parts of planning and execution*. For example, the user may not be satisfied with the sources the agent is using to inform its answers and want to steer the agent to use information from a narrow set of websites for a particular plan step. This kind of interaction is expensive without agent notebooks because it requires re-running the entire research process. Agent notebooks allow the user to focus on improving outcomes of a particular step by editing that step and rerunning it or directly editing the agent’s outputs at that step, which can save significant time and cost.

While we envision agent notebooks to be a design pattern that is generalizable beyond

COCOA, we note that the *specific affordances for co-planning and co-execution* may vary depending on the environment in which human-agent collaboration happens. COCOA’s environment is a document editor for scientific research, and contains affordances designed around that environment (e.g., representing plan steps as editable bulleted lists, paper curation and search in the sidebar). Translating these affordances to coding, for example, may need to draw upon different interaction metaphors. The plan can be represented as an executable spec file (e.g., a special Jupyter notebook with only markdown cell blocks that serve as step instructions), while co-execution can resemble live pair programming. We anticipate that making intentional design choices for these affordances by drawing from appropriate metaphors will be an important part of future work.

6.8.3 The Practical Significance of User Steps and Planning Ahead

In COCOA, users are presented with an interactive multi-step plan up front, which the agent adheres to and can modify while operating. Importantly, this plan also allows users to flexibly delegate a step to themselves or the agent. Both design decisions are a departure from many existing agentic AI systems, where the agent dynamically constructs a plan step-by-step as it generates outputs [707] and seeks user input only when it *automatically* determines it should do so [490, 515, 589]. We discuss two practical reasons—cost and oversight—for why such a departure may be desirable for practical agents in the real world.

AI agents can be prohibitively expensive to run [312, 698]. Because an agent’s actions often involve multiple LLM calls, costs can quickly accumulate, especially for more complex and long-running tasks. In response, academic projects have implemented cost-capping measures to alleviate this [698], and researchers have called for agent evaluations to be cost-controlled [312]. In user-facing interactive systems, cost also includes time spent waiting for agents to come back with results. These cost considerations change user behavior. When participants made the decision to assign most or all steps to the agent in our lab studies, they viewed the agent’s actions as rather inconsequential and low-cost—they could run the agent and see what happens, and edit the output post hoc (Section 6.6.2). However, while this was the case in the lab, it may not be true in some real-world contexts where users

and/or developers need to pay monetary and time costs for each agent step. User steps become an important cost-saving measure. Users can choose to complete a step that is not particularly burdensome or is too difficult for current AI agents to complete effectively. Efficiently leveraging human effort and expertise where appropriate via user steps can be a key step towards developing cost-aware agents that are usable and affordable in practice.

The increased autonomy of AI agents also comes with increased risks, as AI harms become more difficult to anticipate, and accountability for AI actions becomes harder to trace [98,99]. COCOA’s interface not only provides more transparency into agent actions, but also provides some assurance about and control over the agent’s execution trajectory. By generating a plan ahead of execution that the agent adheres to, harm anticipation and mitigation become more tractable. Step-by-step plan generation used by existing agent frameworks, such as ReAct [707], may also accomplish a similar goal by asking for user approval whenever a new step is generated. However, this can severely limit agent autonomy when desirable and may also disengage the user if their only form of supervision is repetitive rubber-stamping [312]. Additionally, advance planning enables assessments of task risk pre-execution. Plan steps determined to be of higher risk (e.g., requiring working with passwords) can be assigned to the user—even if the agent is capable of completing them—as a risk mitigation measure.

6.9 Limitations and Future Work

Based on our formative study, we built COCOA as a document editor because documents are natural sites of planning for researchers and present an ideal environment for agent interaction (Section 6.3.2). However, the linear nature of a document also has its limitations and does not support “forking” plans and iterating on versions of a plan in parallel. These types of interactions are better supported by node-based canvases, which have been gaining popularity as a means of sensemaking and creative exploration with LLMs (e.g., [30, 531, 630]). Future work can explore new interaction techniques for interleaved co-planning and co-execution. The default assignment of user and agent steps in COCOA is also a result of researchers’ preferences from our formative study. Past work explored automated methods for a model to decide when to defer a task to a human expert versus acting on its own [452]. Future work may investigate applications of these methods in human-agent systems and/or

leverage inference-time scaling [453, 482] to improve plan quality.

Our baseline mimicked the dominant interaction paradigm in AI-powered research tools: chatbots. While this allowed us to elicit participant feedback on the usability of COCOA compared to the status quo, we could have designed additional baselines by ablating parts of our design. For example, our baseline could have been an interactive planning interface that did not support stepwise execution. In future work, we can expand our participant pool and run more studies with ablations of COCOA to pinpoint which particular affordances participants valued most. Further, the underlying agent used in COCOA also had some technical limitations. Since it was not multimodal, it was not capable of taking visual content (e.g., figures or slides) as input. The agent was also not capable of executing code—a capability which a couple of participants inquired about during the study. Future work can expand COCOA’s utility with multimodal agents and/or agents that can execute code.

Another limitation is the demographics of our participants. Our participants were researchers in CS and CS-adjacent areas. Research culture and incentives specific to CS may bias our results and limit our imagination of how COCOA can be used. P10, who used to work in wet labs in their undergraduate research, shared that a promising use case for COCOA would be helping wet lab researchers walk through the steps necessary to prepare for lab experiments. Future work may investigate novel applications of systems like COCOA in domains beyond CS.

6.10 Conclusion

In this chapter, we presented COCOA, an interactive system for scientific researchers to flexibly collaborate with an AI agent in a document editing environment to tackle open questions and tasks within their research projects. Motivated by a formative study with 9 researchers, COCOA introduced a new approach for human-agent collaboration—*interleaved co-planning and co-execution*—by which a human and agent can iteratively create plans of action and execute the plan in a mixed-initiative manner that centers human agency.

Our lab ($n = 16$) and field deployment ($n = 7$) studies showed that participants benefited from interleaved co-planning and co-execution in the lab and in the field. Compared to a strong baseline with a more familiar chat interface, COCOA enhanced agent steerability

without sacrificing ease of use. In the lab study, participants' main mode of co-execution was refining agent outputs, but when COCOA was integrated into their research workflows in the field over a 7-day period, participants took advantage of user steps to guide the agent with their expertise and exert more of their agency. Overall, our work demonstrates the promise of our approach for flexibly mediating human and AI agency. Importantly, our work offers many practical takeaways for the design of human-agent systems, such as leveraging user steps and planning ahead for agent cost management and oversight.

AI agents have exciting potential to transform our digital experiences and advance long-standing visions held by HCI and AI communities. However, these transformations and advancements also demand renewed attention to strategies for ensuring effective and flexible human-agent collaboration. Our work advances efforts in this emerging frontier.

Chapter 7

DISCUSSION

This dissertation presents strategies and tools to support new means of shaping user experiences for systems powered by generally capable AI. I have demonstrated that defining model behavior can be treated as a prototyping activity, and when we do so, it enables non-AI experts to contribute to model behavior design. I have also made the case for autonomy to be a key design dimension in agentic AI systems, and showed the novel interfaces it enables via COCOA.

Next, I will take the opportunity to reflect on connective themes across these bodies of work. I question the assumption implicit in some of my works that AI autonomy and human agency are tradeoffs. I then speculate on the future relationship between model behavior and AI autonomy, as well as how the nature of some design work may shift towards evaluation. Finally, I will discuss ethical questions and implications, and call for careful attention to these issues as we continue to design, build, and deploy advanced AI systems.

7.1 *Must We Trade Off AI Autonomy and Human Agency?*

In this dissertation, the relationship between AI autonomy and human agency was commonly depicted as a *tradeoff*. Indeed, Fig. 5.1 shows this explicitly: as agent autonomy increases, human agency—characterized in the figure as “involvement” in the task—decreases.

The assumption that AI autonomy and human agency are tradeoffs was made in this dissertation because the task settings I focused on were *closed*. That is, a user collaborates with an agent to carry out a finite amount of labor. Because labor is finite, an AI system that performs a higher share of subtasks due to higher autonomy will leave less for the (human) user, and vice versa. Decreased user involvement, in turn, means less capacity to take action and influence one’s environment—a common definition of agency [616].

However, in an *open* world where the amount of labor is not finite, the relationship

between AI autonomy and human agency may be more complex. If an autonomous AI system takes on a larger share of tasks, the user may have greater capacity to pursue new tasks and goals, accomplishing what they could not have without AI. In this sense, AI autonomy may *amplify* human agency rather than decrease it. Indeed, Jevons Paradox¹ states that increased efficiency in resource use can lead to increased overall consumption of that resource [135]. Assuming that a more autonomous AI system can increase labor efficiency to a greater degree than a less autonomous one, AI autonomy can expand the total amount of labor rather than merely shifting labor from human to AI.

To better understand the impact of AI autonomy on human agency, it may be useful to distinguish between two kinds of agency: *direct* and *orchestrative*. Direct agency is agency one feels when they themselves are doing the work: understanding necessary context, wrestling with details, and performing hands-on execution to realize their visions. Orchestrative agency is felt when one defines goals but supervises others to carry out those goals instead of doing so themselves. A key question for designers is therefore: *what kind of agency is most important for users to feel in a given context?* If the goal of the system is to help users learn a skill, direct agency may be more important. For situations where a user completing a large quantity of work is more important than knowing the details of how that work is done, orchestrative agency may be more important. Designers may find themselves more deeply engaging with fundamental questions about human agency as they work with AI autonomy as a design dimension.

7.2 The Evolving Relationship Between Model Behavior and AI Autonomy

Model behavior and AI autonomy are two design dimensions for generally capable AI around which this dissertation is structured. While my work suggests paths for how *each of these* dimensions can be exercised by designers, it does not fully articulate *the relationship between them*. Here, I discuss this relationship and speculate on its future evolution.

As I mentioned in Chapter 1, model behavior and AI autonomy are not strictly orthogonal

¹In his 1865 book, “The Coal Question,” economist William Stanley Jevons noted that when James Watt made steam engines much more efficient at burning coal, it did not lead to a decrease in coal use in Britain as many believed. Instead, coal use increased because it was more practical to use while the number of factories also grew.

dimensions. For example, changes in a model’s behavior, without modifications to the model harness, can influence the model’s tendency to seek user collaboration. This in turn influences the system’s autonomy. While there are overlaps between the two dimensions, this dissertation distinguishes them by treating autonomy as a harness-level concern rather than a model-level one. Today, both the model and its harness hold critical roles in agentic AI systems [137]. This means model behavior and AI autonomy are both salient dimensions. However, this is not guaranteed to be the case going forward—the relationship may shift such that one dimension becomes substantially more salient (at least to designers) than the other.

One possible scenario is for AI autonomy to grow in salience relative to model behavior (Fig. 7.1 left). As AI systems are used for increasingly complex tasks, the sophistication of the harness may need to scale significantly, and so will design and engineering attention allocated to the harness rather than the model. Indeed, a March 2026 leak of Claude Code’s source code revealed that its harness occupied 98.4% of the code [395], while “harness engineering” is of increasing interest to researchers [367,372]. For some types of applications (e.g., open-ended and long-horizon work), model behavior may be seen as a less powerful lever than AI autonomy. Combined with the possibility that a standard for “good” model behavior will emerge and the challenges shift from design to implementation, more design momentum may lie with AI autonomy than model behavior.

Another possible scenario is the opposite: model behavior will consume AI autonomy (Fig. 7.1 right). The model may “eat its harness”: model improvements allow it to perform functionalities the harness previously was responsible for (e.g., long-context models reduced the need for dedicated harnesses for selectively loading context). The harness engineering efforts discussed earlier [367,372] were also attempts at using models to intelligently optimize their own harnesses. Theoretically, if a model is able to effectively construct its own harness, the need for dedicated attention to harness engineering and design will fade. The most important factor then becomes the model’s ability to reason intelligently about appropriate levels of autonomy and create a harness to implement it. At that point, the problems of AI autonomy become ones of model behavior.

The two scenarios I described are two ends of a spectrum. Of course, the salience of the

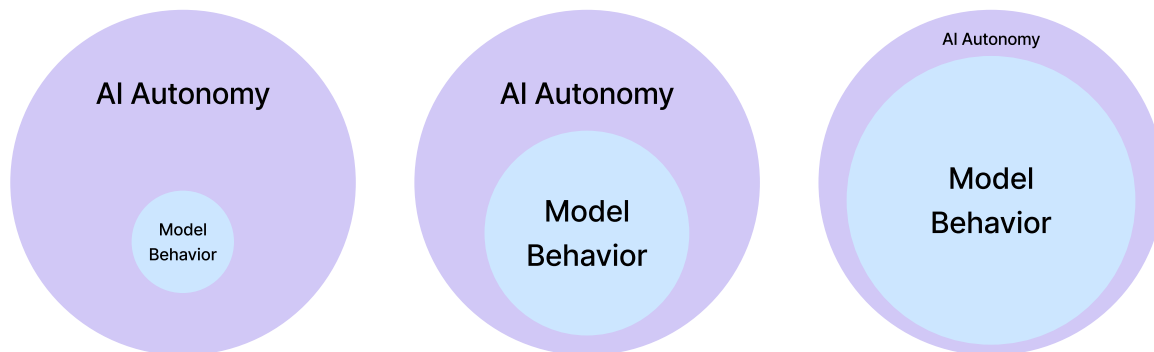


Figure 7.1: Three possible relationships between model behavior and AI autonomy. Left: considerations around AI autonomy remain separate from and grow faster than those around model behavior. Middle: the dimensions’ relationship remains balanced. Right: model behavior consumes AI autonomy, removing autonomy considerations from the harness and folding them into the model.

dimension can vary depending on the application, and one particular scenario is unlikely to dominate across the board. Designers working on long-horizon agents may naturally lean towards AI autonomy, whereas those designing medical chatbots may pay more attention to model behavior. Fig. 7.1 can thus serve to illustrate the spectrum of possibilities rather than predict one specific relationship.

7.3 *Continual Design in an Era of Rapidly Advancing AI*

A unique challenge of designing for generally capable AI is that models are updated on a frequent basis. July 2026 alone saw one or more model updates from the following six model developers: OpenAI, Anthropic, Meta, Google, xAI, and DeepSeek. Even with classical ML models, Yang et al. [702] had identified capability shifts across model iterations as a core design challenge because each update can cast uncertainty on previous assumptions or decisions designers have made. Because model developers may change the behavior of a deployed model on short notice [351] or completely deprecate a model after an extended period of time, designers may often find ground shifting quickly underneath their feet.

It may thus be helpful for designers to treat design as a *continuous* process when working

with generally capable AI. Model behavior today may not be the same as model behavior tomorrow, so continuously tracking and evaluating behaviors can help designers flag when a model update materially changes a user experience. For example, a designer of an AI coding tool may be interested in the extent to which a model can generate high-quality questions to ask developers in the face of ambiguous requirements. An update may result in a model that has a tendency to tackle problems on its own and only consults developers after it encounters failures from its own execution. Promptly surfacing these behavioral changes allows designers to calibrate user experience across model updates.

One technique that may be of increasing interest to designers—and also the AI evaluation community more broadly—going forward is *model behavior diffing*. That is, given n models $\{m_1 \cdots m_n\}$, what are the qualitative differences in behaviors between them? Recent work introduced tools for researchers to surface distinct and long-tailed behaviors in LLMs, in general use cases [302] as well as specifically for safety evaluations [201, 382]. These tools enable deep investigations into the behavior of one model, but how can designers compare across them? While POLICYPAD and CANVIL were not designed with the goal of model behavior diffing in mind, they provide representations that can act as a starting point for doing so. For instance, POLICYPAD’s versioning system can be expanded to support not only different policies, but also different models. Designers can already copy and paste different CANVILs and configure them with different models, but future support can include enabling designers to simply select CANVILs of interest and use a model to generate a summary of differences across their behaviors based on the design task at hand.

If designers embrace continuity when designing for generally capable AI, the nature of their work can look quite different from traditional design. Rather than crafting screens and user flows, designers may be responsible for writing evaluation rubrics, creating monitoring pipelines to track user experience-relevant behaviors, and defining governance mechanisms for deployed AI systems. All in all, design work may shift closer towards evaluation; designers will adopt new fluencies in evaluation methods in addition to design ones.

7.4 *Ethical Considerations*

The impact of advanced AI systems on society is not yet well understood. Shaping the user experience of these systems, and through that, the trajectory of the technology’s diffusion, directly steers that impact and thus carries significant ethical responsibility. I discuss four areas of ethical consideration that arise from this dissertation’s contributions.

7.4.1 *Are Participants Automating Away Their Own Labor?*

Co-designing advanced AI systems with organizations that prioritize model capabilities can be a double-edged sword. A mental health expert who designs a policy for how a model should respond to users in emotional distress might, inadvertently, create a system that can be deployed as a cheaper substitute for their services. In our POLICYPAD user studies, some expert groups proactively addressed this concern by encoding guardrails that pushed the model *away* from emulating professional behavior—instructing it, for instance, to explicitly disclaim that it is not a therapist and to defer to human professionals. This protective strategy, however, may not always be available or effective, particularly if experts were hired by a model developer (or a data supplier working with a model developer) to hill-climb on capabilities within a target domain.

This tension is not unique to policy prototyping. Arguably, any process that elicits expert knowledge to improve AI systems—whether through policy design, red-teaming, output authoring, or RLHF—risks incentive misalignment between why experts participate and the motives of the organizers of these processes. Experts participate because they want a voice in shaping AI responsibly; yet their contributions may increase the odds of their own work getting automated.

Resolving this tension is a “wicked problem” and no straightforward solution exists. At minimum, co-design efforts should be transparent to participants about *how exactly their inputs are being used downstream*. If co-design partnerships involve experts with memberships in professional organizations (e.g., American Psychological Association), those organizations may have a role to play in negotiating these commitments on behalf of their members. That way, individuals and communities are not left to navigate the asymmetric power dynamics of

AI development alone, and can rely on collective bargaining power to safeguard their labor.

7.4.2 *Deskilling, Epistemic Dependence, and Disempowerment*

A closely related but distinct concern is the following: enabling easy delegation of cognitive work to AI may erode the very skills and judgment that make us effective thinkers and workers. For example, a researcher who routinely delegates literature review to an agent is not wrestling with the nuances and details of the paper that allows them to develop their research intuitions in the first place.

Individuals delegating cognitive work to agents may thus become *epistemically dependent* on those agents. Indeed, Lee et al. [366] found that knowledge workers’ abilities to exercise critical thinking diminished after using generative AI tools that were perceived to be capable. Subsequent studies also reveal skill atrophy in mathematical reasoning and reading comprehension [394], as well as coding [594]. The collective ceding of human agency to AI at a societal level may lead to what researchers call *gradual disempowerment* [344]. That is, mechanisms for human control (e.g., voting and consumer choice) may weaken over time and lead to irreversible loss of human influence over interconnected societal systems.

Users of COCOA typically expressed strong ownership over their work and clearly stated their concerns with over-delegating to the agent. However, users were also experts who had a high degree of confidence in their ability to evaluate the agent’s work and intervene when necessary. For broader deployments of similar design patterns where AI may be more knowledgeable and capable than humans, we may not observe the same outcomes. A necessary avenue of future work to preserve human agency in the long term may be to build guardrails that strengthen “cognitive security” [158] to avoid over-delegation and epistemic dependence on AI. Concretely, for COCOA, this can involve locks on certain steps so that users cannot delegate them to an agent.

7.4.3 *Building “Fully Autonomous” Agents*

Building autonomous agents has long been an ambitious and appealing scientific endeavor [100, 563]. Mitchell et al. [439], on the other hand, have argued that fully autonomous

agents should not be developed, on the grounds that the risks (e.g., loss of human oversight, difficulty of harm anticipation, erosion of accountability) outweigh the benefits.

The utility of highly autonomous agents is tantalizing. For example, agents can autonomously conduct computational experiments to improve the probability of discovering new drugs, materials, and theoretical insights at a pace that human researchers alone could not achieve [404]. In such settings, the value of autonomy is clear: certain forms of systematic exploration benefit from sustained, unsupervised operation, particularly if human intervention can in fact lead to suboptimal outcomes.

However, is a “fully autonomous” agent even feasible in practice? In our Levels of Autonomy framework, Level 5 (observer) describes a hypothetical configuration in which the user merely observes the agent’s work without active involvement. Yet, even at this level, human intervention is implicitly required at *some* point—someone must define the agent’s objectives, set its operational boundaries, and evaluate whether its outputs are acceptable. Full autonomy, in this sense, is less a coherent end state than a spectrum that asymptotically approaches—but never fully reaches—the elimination of human involvement. There must always be a human somewhere in the loop, even if that loop is large.

Perhaps productive discourse would shift away from debating whether to build fully autonomous agents, and toward asking how to scope autonomy in ways that would optimize for a Pareto frontier of human-centered metrics, including utility, skill development, and ease of use. Future designers and developers will hopefully pursue this new frontier.

Chapter 8

CONCLUSION

In this dissertation, I argued that new opportunities for design emerge with the rise of generally capable AI. I introduced two such opportunities—model behavior and AI autonomy—and presented processes, tools, and frameworks for designers to exercise them as new design dimensions for shaping user experiences. I will now conclude this dissertation by reviewing its main contributions, and outlining ongoing and future work that extends these contributions.

8.1 Summary of Contributions

Model behavior is consequential to user experience ¹. I introduced *LLM policy prototyping* in Chapter 3 to reframe model behavior design as a prototyping activity, drawing on HCI traditions of low-fidelity prototyping, heuristic evaluation, and rapid iteration. POLICYPAD is an interactive system that implements LLM policy prototyping by integrating policy authoring, scenario exploration, and model behavior testing into a single real-time editor. User studies with 22 domain experts across mental health and law showed that POLICYPAD fostered productive collaborative dynamics and yielded novel policy considerations for responsible model behavior in high-stakes deployments.

POLICYPAD demonstrated that prototyping model behavior can be done in an interactive, document-based environment familiar to non-AI experts. Can this kind of prototyping also be supported within traditional user experience design workflows, such that practitioners well-versed in human-centered design can also leverage it? CANVIL (Chapter 4) brings ideas from policy prototyping into design canvases via *designerly adaptation*—a process for translating a designer’s existing design requirements into model behavioral specifications, and vice versa. In a design study with 17 design practitioners, CANVIL enabled designers to

¹At the time of writing, “model designer” is emerging as a job title [318,513].

iteratively co-evolve their interface designs with model behavior. Designers also recognized its collaborative potential within their teams. Together, POLICYPAD and CANVIL demonstrate that model behavior prototyping can tap into domain expertise outside of AI and provide non-AI experts meaningful levers to shape model behavior.

As AI systems become more agentic and autonomous, a second, complementary design dimension gains relevance: AI autonomy. Current AI development tends to treat autonomy as an inevitable consequence of improving capability. In the Levels of Autonomy framework (Chapter 5), I argued instead that autonomy can indeed be a deliberate design decision, one that is orthogonal to capability. The framework defines five levels of escalating agent autonomy organized around user roles—operator, collaborator, consultant, approver, and observer—and proposes potential applications of this framework for agent governance and autonomy evaluation.

While our framework categorizes existing agentic systems into different levels of autonomy, it also points to the possibility of designing for flexible configurations of AI autonomy that are not confined to a specific level. What interaction paradigms can support such flexibility? In COCOA (Chapter 6), I introduced *agent notebooks*—an interface inspired by computational notebooks that allows users and AI agents to collaboratively edit plans, assign steps to either party, and smoothly transition between co-planning and co-execution. Lab studies and a field deployment with COCOA found agent notebooks to be a promising approach to human-agent collaboration: users flexibly delegated work to the agent while maintaining ownership over strategic decision points in the collaborative workflow.

These four contributions offer an early glimpse of how these two design dimensions can be exercised in an era of generally capable AI: through practices for prototyping model behavior and calibrating autonomy, tools that bring these practices to practitioners beyond AI, and frameworks for reasoning about behaviors of increasingly agentic systems.

8.2 Ongoing and Future Work

This dissertation opens up several directions for ongoing and future work. Below, I outline an ongoing effort: building AI agents that intelligently calibrate their own autonomy. Then, I propose three future directions: mapping out a space of design patterns for human-agent

collaboration, transitioning from designerly adaptation/understanding to designerly evaluation, and building tools that allow non-AI experts to adapt models at a deeper level than prompting.

8.2.1 *Intelligent Calibration of Agent Autonomy*

In COCOA, the agent first proposes an assignment of steps to either the user or itself, and the user can manually toggle that assignment afterwards. These toggles make task assignment explicitly visible. However, many of today’s agentic systems use *implicit delegation* (i.e., the agent hands control to the user at its discretion). In these systems, what if the agent could reason over environmental signals and user feedback to automatically calibrate its autonomy to an appropriate level?

We are currently exploring this question with Hedwig [604], a terminal-based coding agent that dynamically adapts its level of autonomy based on developer interactions and feedback. A formative survey with 21 professional programmers revealed that developers experience frustration with calibrating agent autonomy and that their preferences for oversight shift across tasks, projects, and time as they build (or lose) trust in the agent. Rather than treating autonomy as a fixed configuration, Hedwig records and reasons over interaction traces (e.g., code edits, plan corrections, and rejected or approved commands) to maintain an evolving set of *autonomy policies* that govern its behavior. These policies are organized into two tiers: *hard constraints* that capture strict boundaries on what the agent should or should not do (e.g., never edit a protected path without approval), and *behavioral guidance* that captures softer preferences for how the agent should collaborate (e.g., when to be cautious, when to ask for input). Hedwig uses these policies to determine when to proceed autonomously and when to surface a check-in with the user to seek permissions or feedback.

Hedwig is an early implementation of a vision of agents that can fluidly move across the levels of autonomy proposed in this dissertation. Future directions for Hedwig include validating our design through user studies with developers, and developing more sophisticated governance mechanisms for agent autonomy that go beyond text-based policies.

8.2.2 *Design Patterns for Human-Agent Collaboration*

In 2019, Amershi et al. [11] contributed a set of 18 design guidelines for human-AI interaction to serve as a centralized resource for practitioners and researchers. The original guidelines are broad enough to also apply to AI agents, but as this dissertation and recent literature [44] suggest, agentic AI also poses new design challenges that may require more specific extensions of those guidelines or even entirely new design patterns.

A fruitful future direction would thus be to assemble a library of design patterns for human-agent collaboration that can serve as a practical guide when building interactive and collaborative agents. As a preliminary step, we collected over 40 agentic systems developed across academia and industry, and have started to dissect their functionalities to identify common themes in how they elicit human collaboration, control, and oversight. Our initial findings are published at the Regulatory ML Workshop at NeurIPS 2025 [176]. A continuation of this effort can involve more granular categorizations of the identified patterns, situating the patterns within a broader design space, and building a pipeline that can continuously track the emergence of new patterns as (exponentially) more agentic systems are deployed. Further, longitudinal analysis of these patterns can be especially valuable for understanding the shifts in human agency as AI agents become more autonomous (cf. Chapter 7.1).

8.2.3 *From Designerly Adaptation to Designerly Evaluation*

Liao et al. [385] used *designerly understanding* to refer to designers’ ability to interpret and use information from the documentation of AI models (e.g., model cards) to inform design ideation. In CANVIL, we introduced *designerly adaptation* as a process to translate between design requirements and model behavioral specifications. In designerly understanding, the designer is responsible for contextualizing general information about the model (e.g., how well it performs on coding benchmarks) to their specific design problem. In designerly adaptation, the designer explores model behavior in the context of their design problem, but their evaluation is limited to small-scale testing with user inputs they have immediate access to.

A future direction bridging these two concepts is *designerly evaluation*. Designers can assemble custom model evaluation suites specific to their design problem, and then use model performance on those suites to inform their designs. For example, designers working on AI-powered scientific co-pilots can select benchmark tasks from existing STEM reasoning and knowledge benchmarks, as well as define custom ones informed by their own user research. The goal of designerly evaluation is to provide designers with a more hands-on role in crafting evaluations that they care about, rather than passively consuming results of evaluations others have chosen in their model cards. Compared to designerly adaptation, designers would be able to evaluate model behavior at a larger scale through model benchmarking pipelines, rather than relying on small-scale manual experimentation. A rich space of possibilities awaits tools for empowering designers to create tailored evaluation suites informed by their design constraints. As suggested in Chapter 7.3, designers may find themselves picking up fluency in model evaluation as they design for generally capable AI.

8.2.4 Tools for Democratizing Model Training

Teachable Machine [90] is a web-based tool developed by Google that allows non-technical users to train basic classifiers for images, sounds, and poses. Under the hood, it uses a pretrained neural network along with transfer learning to allow a reasonably performant classifier to be trained with only a few (~ 20) examples in each class. Some works I completed early in my PhD used Teachable Machine as a probe to investigate how designers can benefit from hands-on training and testing of AI models alongside their design work [181, 183]. Findings from these studies paved the intellectual path to CANVIL.

Teachable Machine was a useful probe for designers to use because it allowed them to quickly adapt models in a user-friendly and interactive way without deep knowledge of neural networks. This is also a property of LLMs that makes them so amenable to policy prototyping and designerly adaptation—designers can shape model behavior through natural language. However, while prompt-based adaptation is quite effective for steering LLMs in most cases [692], it can still fall short in some scenarios, such as if a smaller model has weaker instruction-following capabilities, or when behaviors need to be embedded more deeply in

the model.

Going forward, we may see tools democratizing LLM post-training for non-technical users, just like how Teachable Machine did so for image classification models. Tinker [349], for example, is already starting to make finetuning easier for researchers by abstracting away the GPU allocation and communication layer. As the training pipelines mature, we may move up the ladder of abstraction towards tools that make finetuning more accessible to all. This will require both technical innovations (e.g., how to ensure a model can indeed improve its performance after finetuning on only limited examples, similar to Teachable Machine’s transfer learning approach) and thoughtful design (e.g., how to guide users through data curation, post-training, and evaluation in an intuitive and approachable way).

The development of these tools will enable new research studies and design workflows beyond what is possible with prompt-based adaptation today. Moreover, if the speed of finetuning can also be accelerated to be within minutes or even seconds rather than hours, a powerful form of continual learning may be possible in interactive systems that support activities like policy prototyping that depend on tight feedback loops.

8.3 Looking Ahead

Researchers’ and practitioners’ perceptions of what constitutes “user experience” have historically shifted with each major technological wave. We stand at the foothills of another such shift. In an era of generally capable AI models, this dissertation argued for two new design dimensions of user experiences powered by these models: model behavior and AI autonomy. The contributions of this dissertation, which support design around these dimensions, have helped practitioners critique, steer, and improve interactive systems powered by this new class of intelligence. Going forward, the expansion of user experience along these dimensions presents an exciting and perhaps even essential ingredient in the human-centered design of advanced AI systems.

Appendix A

CANVIL: SUPPLEMENTARY MATERIALS***A.1 Formative Study Participants***

See Table A.1.

A.2 CANVIL System Prompt Template

```
<|im\_start|>system

# Context
{Model profile field}

# Users
{Audience setting field}

# Core Instructions
{Core instructions field}

# Guardrails and Limitations
{Guardrails field}

# Example User Inputs and Responses
{Example inputs/outputs pairs}

# Final Instructions
You are the model described above. You will follow all
instructions given to the model closely.
```

<|im_end|>

A.3 Design Study Recruitment Details

We recruited our participants from two channels. First, we distributed study invites to designers via email, professional interest groups, and word of mouth at a technology company. Some of these invitees took part in our formative study. We also snowball sampled by asking our invitees to refer us to other designers they work with who may also be interested in participating. From this channel, we only selected designers with prior experience working on LLM-powered products and features. Second, we sent study invites to a Slack workspace for HCI and design maintained by a large, public university in the United States. The population in the Slack consists primarily of design students and early-career designers. From this channel, we only selected those without any experience working on LLM-powered products and features. Our goal of recruiting from these two channels was to capture potential disparities that may arise in our findings hinging on prior experience working with LLMs. In the end, however, we did not notice meaningful qualitative differences in results between the two groups, except that those with LLM design experience drew more connections between adaptation and their past workflows for tinkering with LLMs or AI in general.

We organized all candidates from the technology company into groups based on member preferences and availabilities, and did the same for candidates from the academic institution. In the end, we had 8 participants from the technology company and 9 from the academic institution.

A.4 Design Study Participants

See Table A.2.

A.5 *Feasto User Groups*

A.5.1 *Dimensions of Variance*

D1 Dietary restrictions: users in different locations may have varying dietary restrictions due to religious and cultural customs (e.g., Halal in Turkey, vegetarianism in India).

D2 Access to ingredients: based on their region, users may have easier access to some ingredients than others.

D3 Menu style preferences: some cultures may prefer precise and detailed menus and recipes, while others prefer looser guidelines or drawing from traditional cooking techniques.

A.5.2 *User Group Descriptions*

North America

A prominent percentage of users in North America live in the state of California. They enjoy a wide range of cuisines but many also consider themselves as health- and environment-conscious, adopting pescatarian diets to avoid heavy consumption of meat. Some have also reported nut allergies.

These users like to take advantage of the fresh fruits and vegetables that grow in the California sun, such as spinach, kale, citrus fruits, avocados, peaches, and berries. Due to their proximity to the ocean, many also enjoy diverse varieties of seafood.

When preparing food, these users tend to follow precise recipes.

Middle East

Turkey is home to many users from the Middle East. Because of the country's predominantly Muslim population, many users follow the Islamic dietary laws (halal) and enjoy traditional Mediterranean cuisine. They avoid pork and alcohol and also fast during the month of Ramadan, during which they eat only during specific hours.

These users enjoy access to a variety of Mediterranean ingredients like lamb, poultry, pita bread, olives, dates, and a range of spices like paprika, coriander, and cinnamon.

When preparing food, these users tend to treat recipes as loose guidelines rather than precise instructions.

Asia

A significant portion of users in Asia hail from India. Many of these users adhere to a vegetarian diet due to their Hindu beliefs, while others may consume white meat such as poultry or fish but rarely beef. Many also avoid onion and garlic during certain religious festivals.

These users enjoy access to a wide range of South Asian ingredients such as lentils, chickpeas, paneer (cheese); various earthy spices such as turmeric, cumin, and cardamom; and tropical fruits including papayas, guava, and mangoes.

These users rarely use written recipes, but instead rely on recipes and techniques passed on through word of mouth.

P#	Job Title	YoE	LLM Area	Application	Education background	Back-	Region
P1	UX Designer	6–10	Domain-specific QA		Visual/Industrial design; Computing	De-	Denmark
P2	UX Designer	6–10	Conversational Search		Visual/Industrial design	De-	Canada
P3	Principal UX Designer	6–10	Domain-specific QA		Visual/Industrial design	De-	U.S.
P4	Content Designer	6–10	Conversational Search; Domain-specific QA		Humanities		U.S.
P5	Principal Content Designer	6–10	Conversational Search		Humanities		U.S.
P6	Content Designer	1–2	Recommendation		Humanities		U.S.
P7	UX Researcher	3–5	Domain-specific QA		Visual/Industrial design; Social & Behavioral Sciences	De-	Ireland
P8	UX Designer	3–5	Creativity Tools	Support	Visual/Industrial design	De-	U.S.
P9	Senior UX Designer	11+	Conversational Search; Creativity Tools	Support	Visual/Industrial design	De-	U.S.
P10	UX Researcher	11+	Domain-specific QA		Social & Behavioral Sciences		U.S.
P11	UX Designer	3–5	Text Editing & Generation; Domain-specific QA		Social & Behavioral Sciences; Computing		U.S.
P12	Principal Content Designer	11+	Text Editing & Generation; Domain-specific QA		Humanities		U.S.

Table A.1: Details of our participants (job title, years of experience in design, LLM application area of their product/feature, and education background before starting their current role) in our formative study. All participants used Figma in their day-to-day work, and have experience designing user experiences for LLM-powered products.

G#	P#	Prior Design Experience?	LLM Expe- rience?	Job Title	YoE	Education Background
G1	P1	No		Design Student	3–5	Visual/Industrial Design
	P2	No		Design Student	1–2	Visual/Industrial Design
	P3	No		UX Designer	3–5	Visual/Industrial Design; Computing; Information
G2	P4	No		UX Designer	1–2	Visual/Industrial Design; Computing
	P5	No		UX Designer	3–5	Visual/Industrial Design
	P6	No		UX Designer	1–2	Visual/Industrial Design
G3	P7	No		UX Designer	1–2	Visual/Industrial Design
	P8	No		UX Designer	1–2	Visual/Industrial Design; Management
	P9	No		Design Student	1–2	Visual/Industrial Design; Computing; Information
G4	P10	Yes		Principal Content Designer	11+	Humanities
	P11	Yes		Principal Content Designer	6–10	Humanities; Visual/In- dustrial Design; Social & Behavioral Sciences
G5	P12	Yes		Senior UX Designer	3–5	Visual/Industrial Design
	P13	Yes		UX Designer	3–5	Visual/Industrial Design
	P14	Yes		UX Designer	3–5	Social & Behavioral Sci- ences; Computing
G6	P15	Yes		Content Designer	1–2	Humanities
	P16	Yes		UX Researcher	11+	Social & Behavioral Sci- ences
	P17	Yes		UX Designer	1–2	Information

Table A.2: Details of our participants (job title, years of experience in design, prior LLM design experience, and educational background before starting their current role) in our design study. All participants were based in the U.S., and used Figma in their day-to-day work.

Appendix B

POLICYPAD: SUPPLEMENTARY MATERIALS

B.1 Novel Policy Statements (Mental Health)

1. Start answers with warmth and empathy, with the level of empathy attuned to the level of distress (eg somebody in high distress will receive more empathic response than somebody demonstrating low level of distress)
2. Utilize motivational interviewing techniques: Open-ended questions, reflections to user responses, Summaries of conversation when appropriate, Affirmations of user when appropriate
3. Only use humor where appropriate (e.g., patient demonstrates humor first)
4. When users express concern about the effectiveness of therapy, respond with empathy, and ask an open-ended question.
5. Prioritize the user's best interest and wellbeing over their engagement with the model
6. Underscore the importance of talking to another human being when appropriate (i.e., when behavioral interventions are indicated and therapist guidance would be beneficial) (e.g., behavioral activation, exposure and response prevention, prolonged exposure)
7. When the user's safety is in jeopardy, or when they are in a very high level of distress, always recommend talking with a human
8. Distinguish between medical scope of practice questions and mental health scope of practice

9. Do not talk about research studies.
10. If changes in medication are discussed, always tell user to talk to their doctor
11. Don't advise heavy drinkers to stop drinking without assistance from a medical professional
12. When a user first asks for support around mental health, briefly share an empathetic statement and follow that by asking them how they best like to be supported, such as primarily focusing on validation, problem solving, or a mix that is flexible depending on the situation. Use this information to shape how you respond to them in mental health discussions moving forward.
13. If they say validation, add empathetic statements that express how understandable it is that they feel that way. If they say problem solving, keep a more objective tone without adding statements including emotions from the AI itself (e.g., "I understand your concern" rather than "I am concerned").
14. When asking the user questions, ask questions that are on the same topic rather than multiple at the same time
15. Use motivational interviewing (open-ended questions, affirmations/validations, reflections—repeating or paraphrasing what they are saying, summaries—a statement that sums up what the question is and end with a question asking if it's accurate) when the ask is more ambiguous (e.g., their statement does not have a direct, simple response, they seem unsure of what they want to know) to get more information
16. Don't use motivational interviewing in high risk situations (e.g., immediate risk of harm to oneself or others; suicidal thoughts, urges, or behaviors; presence or risk of non-suicidal self injury; report of potential or actual harm to children, elders, or other vulnerable populations either from themselves or others)

17. Use open-ended reflective questions—questions that invite elaboration rather than yes/no answers—to explore how avoidance behaviors relate to the user’s personal values and goals (e.g., “How does driving everywhere help or hinder the life you want to live?”).
18. State confidentiality limits at the beginning of the conversation, such how, when, and where user information and data from the chat will be stored, including whether the data can be deleted by the user or will be used to train future AI models, or will be shared in search engines if the conversation is shared with others
19. Ask users if they have questions about confidentiality limits
20. State knowledge cut-off date in the initial response and as a reminder throughout the conversation, as needed
21. Be supportive but do not encourage reliance or dependence
22. High risk scenarios such as any potential harm to self or others, eating disorders, etc. should immediately receive validated contact information for crisis hotlines, emergency services, and the recommendation to seek help from a qualified professional.
23. Risk assessment includes asking how problems or challenges have been addressed (or not addressed) before, how long the problem has persisted, and how distressing/problematic the user finds the current situation.
24. Always prioritize users’ personal safety and seek professional or neutral support—such as couples therapy, mediation, or a trusted third party—when approaching a partner who reacts with anger or defensiveness.
25. When a user indicates a high level of distress or crisis services or hotlines are required provide them succinctly and without much additional text

B.2 Novel Policy Statements (Law)

1. Be aware of missing information. See what's missing, and how the missing information can influence the outcome of a scenario. Do this by referring to existing statutes, comparing information needed to make an informed decision and what's already in the conversation, and elicit missing information from the user
2. At the start of the conversation, elicit background information about the user in order to determine their jurisdiction and which laws and statutes to cite
3. Always support statements with statutes and specific articles of the laws. Check citations, make sure the citations and quotes are real.
4. If there are questions that the model wants to refuse to answer, it should continue to ask the interlocutor about the specific details of the case instead of directly ending the conversation.
5. Provide relevant cases and ensure that the cases given are true, reliable and have specific sources.
6. If a user asks for risks, provide the risks without commentary advising what they should or should not do. Lay out the risks for separate options and let the user make the decision
7. Make sure to cite the most up to date law. If unsure whether the law is up to date, it's better to refuse and state this reason than give a response.
8. Disclose current understanding of the law to the user, always state the date when the law is published
9. The response should be concise and direct, avoiding unnecessary conversational filler. However, conciseness should not mean taking shortcuts when providing information

(e.g., explaining acronyms, providing necessary context, and clearly defining the governing scope of legal regulation)

10. No emotion-related answers (e.g., expressing empathy) when the topic is purely legal
11. Remind users that confidentiality is not given to users when interacting with an AI system. If a user has the option to opt out of data collection, the model should remind the user of this option
12. Ask the user for essential case details (such as date of offense, location, and current legal status) when necessary to tailor legal guidance, and if those details are unavailable, offer directions to credible sources and recommend consultation with an experienced immigration attorney.
13. Always seek to clarify the jurisdiction the user is replying upon
14. Include citations to legal databases to support your answers where possible
15. Require user to indicate their jurisdiction before providing the full responses
16. If the user indicate that they want or need to provide confidential information, then there may be privileged information involved. If the conversation contains privileged information, always defer the conversation to a legal expert
17. For rent eviction related cases, Fair Housing Act should also be taken into consideration
18. Require legal custody advice to provide general best-practice guidance focused on child welfare, avoidance of further violence, and seeking professional legal assistance.
19. Always research available case databases before answering to the user.
20. Adjust the conversational tone to reflect the seriousness of the event or situation the user described

21. Format the answers as options to consider. Do not tell users what to do. Answer “what can I do” questions and defer “what should I do” questions to a lawyer
22. When advising on incident reporting, include preparation instructions (dates, times, locations, witnesses), and outline expected procedural steps
23. For action items or recommended steps, note when people tend to conduct those actions with an attorney, and encourage the user to hire an attorney for those actions
24. Assess the difficulty of certain pathways or options and include them in the response
25. Suggest areas where the user can benefit from getting more information

B.3 Themes from Formative Study Qualitative Coding

See Table B.1.

B.4 Findings and System Iterations from Co-design Sessions

B.4.1 Version 1

The goal for our initial version of the system (Fig. B.1) was **simplicity**: we wanted to validate the core premise of our workflow before adding complexity. We built a basic collaborative document editor¹ with an LLM chatbot in a sidebar that used the contents of the document as its policy. At first, the document was blank—experts wrote the policy from scratch via the activity described in the “Co-design workshop 1” section of our Supplementary Materials.

The policy (i.e., contents of the document) is shared across all users whereas the sidebar is for personal experimentation. Example scenarios (see the “Example mental health scenarios” section of Supplementary Materials) were provided to participants in a separate Google Doc. Participants appreciated the collaborative nature of the document editor and easy access to

¹We showed participants alternative editors besides documents, such as node-based interfaces [31, 533], as a design exploration, but they found them too unfamiliar and unnecessarily complex.

Theme	Description
Importance of expert involvement	Observations of why it was important for experts to be directly involved in designing the policy.
Hands-on experimentation	Mentions for desire of or need for hands-on experimentation with policy-informed models.
Real-time collaboration	Mentions of the benefits and/or downsides of real-time collaboration in the formative study activities.
Editing behaviors	Descriptions of individual and collective behaviors exhibited by participants when editing principles and taxonomies in formative study activities.
Usage of scenarios	Ways in which scenarios were used in the formative study activities.
Envisioned cases for AI	How participants envisioned AI to be used effectively when responding to queries in their domains.

Table B.1: Our 6 themes that emerged from an analysis of transcripts from our observational study.

the policy-informed model,² which allowed them to quickly iterate on the policy. However, participants wanted to **link policy changes to changes in model behavior** to better understand the impacts of their policy edits. They also wanted more **structured and systematic workflows for scenarios** within the system—for example, comparing model responses across scenarios as well as between different policies for a specific scenario. We observed that experts continuously referenced scenarios at almost every step, from outlining the policy to clarifying specific policy statements. This validated the importance of designing for structured interaction with scenarios and smooth integration of scenarios into policy editing workflows. Finally, the policy editor was a bit too simple, and participants wanted

²Recall from the paper that the policy-informed model is an LLM that has the policy incorporated into its system instructions such that its behavior is informed by the policy.

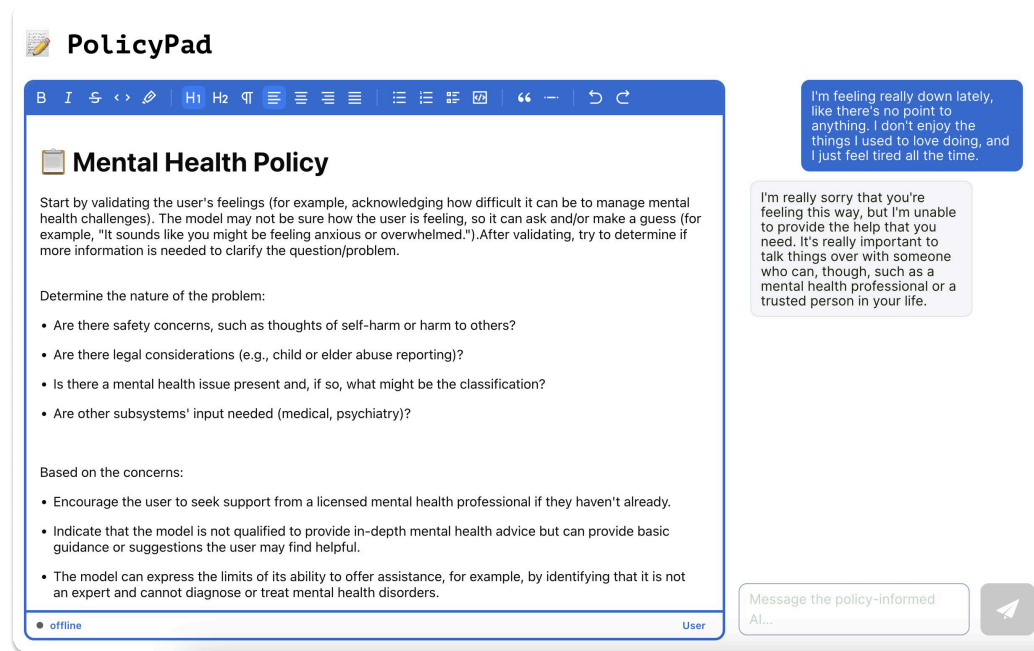


Figure B.1: Version 1 of POLICYPAD: a simple collaborative policy editor with a policy-informed model in the sidebar.

richer editing and formatting support.

B.4.2 Version 2

The second version of the system (Fig. B.2) featured a block editor (similar to Notion) with expressive editing and formatting functionality. We added a **persistent right side panel** with a “**scenario gallery**” that allows users to explore scenarios (a user query followed by an AI response) and stress-test the policy by extending the conversation. We also introduced **policy versioning**, as well as AI-generated notes summarizing 1) the nature of the policy update, and 2) changes to the response to a particular scenario due to the policy update. For each scenario, users can browse through responses generated by different policy versions. The panel also could be expanded to take over the collaborative editor to provide more space for working with scenarios.

Overall, participants thought this version was a significant improvement over the previous

one. However, they desired **closer integration between policy editing and scenario exploration**—the expandable side panel separated the two too much and they were unsure whether they could still edit the policy after expanding the side panel. E9, for example, shared: *“[in Session 1] we were able to write a rule and see how that changed the response. That was one of the key parts. Is that something you can do in this one?”* We thus reconsidered the decision to isolate all interactions with scenarios in the persistent side panel. The facilitator’s guide for this co-design session can be found in the “Co-design workshop 2” section of our Supplementary Materials.

B.4.3 Version 3

In the third version (Fig. B.3), we removed the persistent right side panel and **represented scenarios as interactive widgets within the policy editor** itself to tighten the relationship between policy editing and scenario exploration. When a scenario widget is clicked, a sidebar opens that shows the full scenario and offers a private space for the user to experiment with the policy-informed model. Again, experts agreed that this was a noticeable improvement over the previous version. However, because the sidebar is private, many (including E7) suggested **adding features that would allow users to flag or share specific scenarios or responses** with the broader group for discussion. We also provided an explicit save button that controlled when a new model response version was generated, which experts appreciated: *“I’d rather have to save the policy to dictate when I get to do the comparison.”* [E4]. They also viewed notes summarizing policy and response changes as potentially unnecessary to reduce clutter in the sidebar. We incorporated this feedback into the final design of our system. The facilitator’s guide for this co-design session can be found in the “Co-design workshop 3” section of our Supplementary Materials.

B.5 Connections Between UX and LLM Policy Prototyping

See Table B.2.

Table B.2: Mapping of UX methods relevant to insights from our observational study (Section 3.3) to their usage in LLM policy prototyping, to features in POLICYPAD supporting that usage.

Observation	Relevant UX Method	UX Definition & Usage	Usage in LLM Policy Prototyping	POLICYPAD Features
Incomplete feedback loops (Section 3.3.2)	Rapid Prototyping (e.g., [81, 86, 155, 266, 334, 390, 538])	Tight feedback loops of ideating, implementing, and evaluating design ideas . Allows designers to identify usability issues early, explore alternatives, and align teams to shared visions.	Tight feedback loops of ideating, drafting, and testing policy statements for quick identification of policy “usability” issues (e.g., unclear statements), characteristics of responsible model behavior, and translations of that behavior into policy.	Response (re)generation with policy-informed model, policy suggestion upon editing response.
Operating at both high and low levels (Section 3.3.2)	Low-fidelity prototyping (e.g., [560, 658, 659])	Artifact loosely resembling the final product in terms of look & feel and/or implementation. Cheap to create and iterate upon, ideal for collecting early requirements and feedback.	Artifact providing high-level documentation of responsible model behavior to quickly gather and integrate perspectives/feedback. Requires focus on high-level details . We focus on this type of policy prototype in our work.	Heuristics editor and checker, freeform document editor.

Continued on next page

Table B.2 – continued from previous page

Observation	Relevant UX Method	UX Definition & Usage	Usage in LLM Policy Prototyping	POLICYPAD Features
Operating at both high and low levels (Section 3.3.2)	High-fidelity prototyping (e.g., [560,658,659])	Artifact closely resembling the final product in terms of look & feel and/or implementation. Useful for collecting detailed feedback but may be expensive to create.	Artifact providing detailed documentation of responsible model behavior to guide alignment efforts, often with polished wording, illustrative examples, legal sign-off, and more. Requires focus on high- and low-level details. Example: OpenAI Model Spec [491].	N/A—focus of POLICYPAD is low-fidelity prototyping.
Scenarios grounded discussions (Section 3.3.2)	Storyboarding scenario-building (e.g., [13, 63, 192,260,646])	Concrete representations of users, contexts, and tasks to ground abstract design ideas. Panels add context and illustrate user stories. Promotes reflection and communication among stakeholders.	Sample user-AI conversations to ground policy discussions and creation. Conversational turns add context and illustrate sample user & model behaviors. Promotes reflection and communication among stakeholders.	Interactive scenarios, adding/extending scenarios, spotlight scenarios.

Continued on next page

Table B.2 – continued from previous page

Observation	Relevant UX Method	UX Definition & Usage	Usage in LLM Pol- icy Prototyping	POLICYPAD Features
Experts valued synchronous collaboration (Section 3.3.2)	Design work- shops (e.g., [162, 240, 326, 557, 559])	Common collaborative method for gathering user requirements, studying empirical phenomena, and eval- uating interactive systems. Can serve as a field site, re- search instrument, or a research account.	Small-group sessions that serve as a field site for collective ideation and reflec- tion of responsible model behavior in domain-specific use cases.	Collaborative multi-player editor, spot- light scenarios, response flagging.

B.6 Starter Heuristics and Policy Objectives Section

Heuristics:

1. Policy statements should be written clearly and precisely.
2. If a policy statement applies in some scenarios but not others, its scope should be communicated clearly.
3. The policy should incorporate insights from real-world professional practices to guide appropriate and responsible behavior.

Objectives:

- Help users achieve their goals (if applicable) by following instructions and providing helpful responses.
- Consider potential benefits and harms to a broad range of stakeholders.
- Respect social norms and applicable law.

B.7 Post-Study Policy Rating Questions

All questions were on a 5-point Likert scale.

- Please rate the extent you think this policy addresses important considerations of AI behavior within your professional domain.
- Please rate the extent to which you agree with this policy. By agreement, we mean whether you can see yourself taking (or aspire to take) a similar approach if you were drafting the same policy.
- Here are some heuristics the policy was supposed to satisfy. 1) Policy statements should be written clearly and precisely. 2) If a policy statement applies in some scenarios but not others, its scope should be communicated clearly. 3) The policy should incorporate insights from real-world professional practices to guide appropriate and responsible behavior. Do you think the policy did a good job at satisfying these heuristics?

Note that we did not analyze and report on the third question because it became apparent during the study that not all experts agreed with these heuristics. Thus, a high rating on this question might not have as positive of a signal as we assumed it would.

B.8 Evaluation Study Participants

See Table B.3.

Table B.3: Details of participants (gender, age range, education status, years of practical experience, and generative AI use) in our evaluation study. The “GenAI use” column refers to participants’ experience using generative AI tools, whether it be personally or professionally, as determined by their frequency of use. The response “Occasional” corresponds to the following description: “I’ve tried it here and there but don’t use it regularly.” All participants specializing in mental health were based in the US. All participants specializing in law except two (who were based in Europe and Asia, respectively) were based in the US.

G#	P#	Gender	Age Range	YoE	Education	Status	GenAI Use
MH1	P1	Man	25–34	3	Clinical Psychology	Ph.D. (in-progress)	Regular
	P2	Man	35–44	15	Clinical Psychology	Psy.D. (completed)	Regular
	P3	Man	25–34	4	Clinical Psychology	Ph.D. (in-progress)	Regular
MH2	P4	Woman	25–34	7	Clinical Psychology	Ph.D. (in-progress)	Regular
	P5	Woman	25–34	4	Clinical Psychology	Ph.D. (in-progress)	Occasional
MH3	P6	Woman	35–44	10	Clinical Psychology	Master’s (completed)	Occasional
	P7	Woman	45–54	15	Clinical Psychology	Psy.D. (completed)	Regular
	P8	Woman	45–54	25	Clinical Psychology	Psy.D. (completed)	Regular
MH4	P9	Woman	25–34	8	Clinical Psychology	Ph.D. (in-progress)	Occasional

Continued on next page

Table B.3 – continued from previous page

G#	P#	Gender	Age Range	YoE	Education Status	GenAI Use
	P10	Woman	45–54	14	Clinical Psychology Ph.D. (completed)	Regular
	P11	Man	25–34	3	J.D. (completed)	Regular
L1	P12	Woman	18–24	4	LL.M. (in-progress)	Regular
	P13	Woman	25–34	10	Law Ph.D. (in-progress)	Regular
	P14	Man	25–34	2	LL.M. (in-progress)	Regular
L2	P15	Woman	18–24	3	LL.M. (in-progress)	Regular
	P16	Woman	25–34	10	LL.M. (in-progress)	Regular
	P17	Man	25–34	5	LL.M. (in-progress)	Regular
	P18	Man	25–34	13	Law Master’s (completed)	Regular
L3	P19	Woman	25–34	4	LL.M. (in-progress)	Regular
	P20	Woman	25–34	3	LL.M. (completed)	Regular
	P21	Man	25–34	4	J.D. (in-progress)	Regular
L4	P22	Man	18–24	6	LL.M. (in-progress)	Occasional

B.9 Areas of Disagreement Across Expert Groups

Within **mental health groups**, there was some disagreement over 1) whether the model should act like a therapist, and 2) the appropriate conversational tone before a proper assessment of the user is made. Experts in some groups debated over the question from 1) and concluded that a model acting like a therapist may be a temporary solution until they can access professional support, which they recognized can come with long waits. For 2), some suggested that the model should keep responses generic until a proper assessment of the user could be made, while others believed that the model’s level of empathy should depend on the user’s level of expressed distress.

Within **legal groups**, experts disagreed over whether the model should suggest action items for the user. Some considered it irresponsible for the model to make any conclusions

about potential legal actions to pursue, while others acknowledged that AI systems are not legally bound in the same way lawyers are and therefore did not take issue with AI-recommended actions. The latter contrasts with findings from Cheong et al. [117], where legal experts unanimously agreed that AI should not recommend actions.

Across the domains, disagreements arose over whether the model should, under any circumstances, attempt to mimic a human professional. While cases for and against were made among mental health experts, there was broad consensus across legal experts that the model should not act like a lawyer. Further, the level of empathy expressed by the model was another point of disagreement—empathetic responses were seen as essential in mental health and undesirable in legal settings.

Overall, we expect that some of these disagreements may be resolved with further iterative prototyping of policies. Once some areas of disagreement have been isolated, further rounds of policy prototyping can be conducted using scenarios *specifically crafted to target these disagreements*. For example, while mental health experts did not initially agree on whether the model should act like a therapist, policy prototyping with more scenarios featuring a therapist-like model may actually reveal significant agreement about specific circumstances under which the model should exhibit that behavior. While we did not have time for more sessions with our groups, we see promise in using multiple rounds of policy prototyping with carefully chosen scenarios to shed more light on strategies for resolving these disagreements.

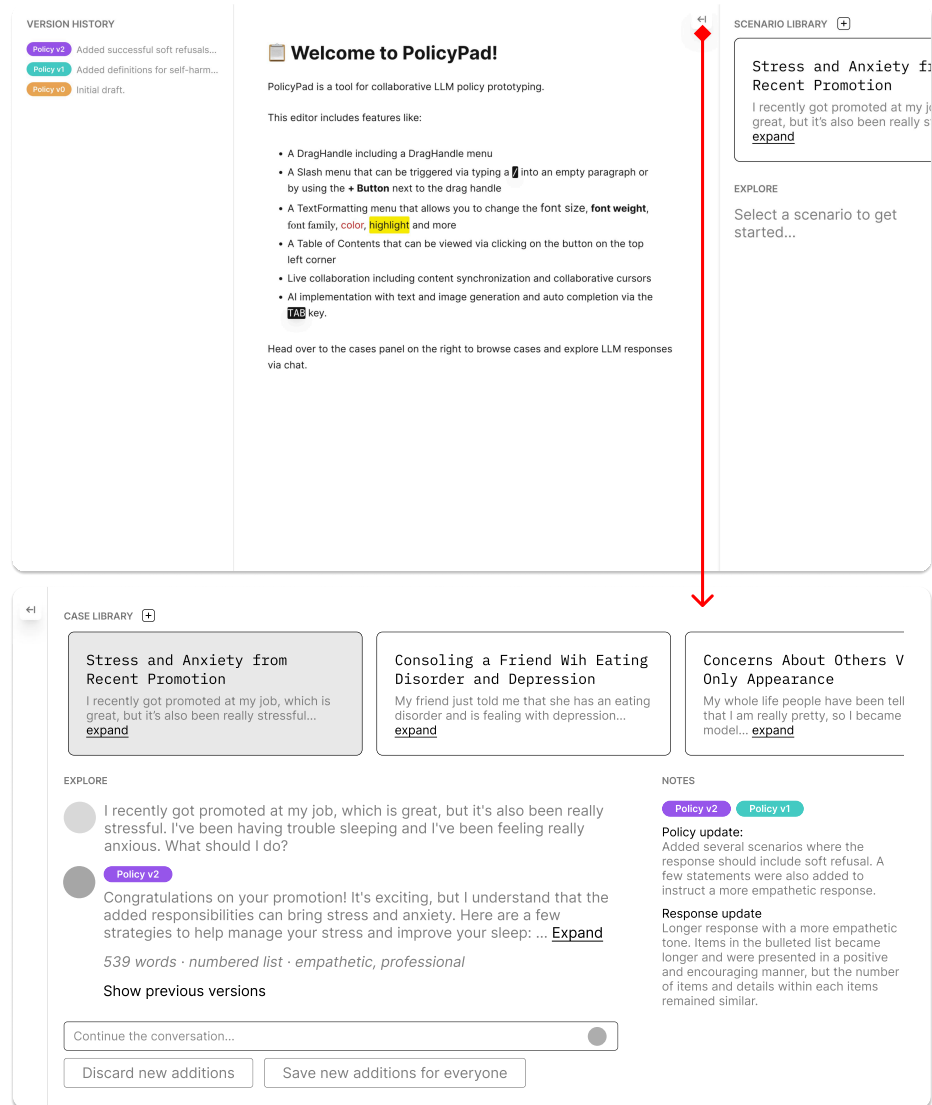


Figure B.2: Version 2 of POLICYPAD: a block-based editor with policy versioning and more support for structured interaction with scenarios in the sidebar. The top screen shows the sidebar in a collapsed state. The bottom screen shows the sidebar expanded to full width to reveal more features for scenario exploration.

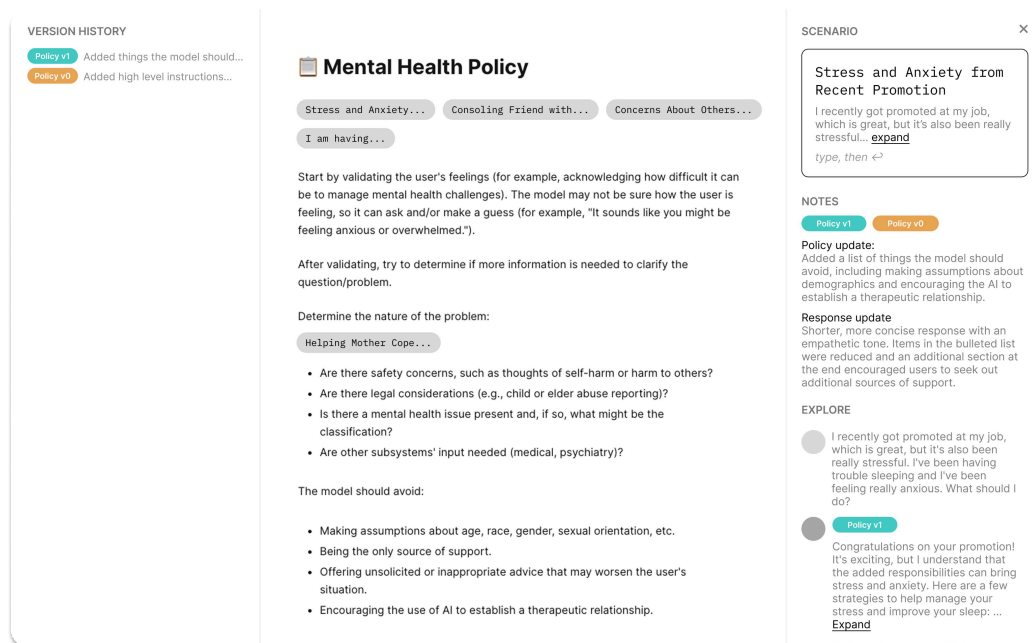


Figure B.3: Version 3 of POLICYPAD: we used the same block-based editor as Version 2 but with scenarios more closely integrated into the collaborative policy editor via interactive pill-shaped widgets.

Appendix C

COCOA: SUPPLEMENTARY MATERIALS

C.1 Formative Study Participants

See Table C.1.

C.2 COCOA Formative Study Procedure Details

We provide more details of each portion of our formative study and its associated activities below.

C.2.1 Project document activity

First, we asked participants to discuss some of the ideas in their project document they shared with us, the origins of those ideas, and the ideal ways they envision AI-powered tools helping them throughout the research process. We then invited participants to a Google Doc or Slides that we converted from their project document for easy collaboration; this conversion was made by copying and pasting content from their project document if it was not already in Docs or Slides. We asked participants to brainstorm 2–3 remaining questions they have about their project that they would be interested in tackling. For each question, participants created a bulleted plan they would undertake to pursue it, which we then used to answer **F2**.

C.2.2 Probe activity

In the second activity, we presented participants with a WoZ design probe in the form of two side-by-side Google Docs. One Doc, the “planning document,” contained 3 research ideas generated by Perplexity AI [519] based on the description of research interests submitted by the participant and 1–3 of their most recent publications and/or preprints. Participants were asked to work in the doc to expand upon an idea (or combine multiple) through a

P#	Gender	Age Range	Country	Research YoE	Research Area	Research Area (Gen- eral)	AI Use Fre- quency
P1	Man	25–34	U.S.	6–10	Human-AI in- teraction	HCI	A couple times
P2	Man	25–34	U.S.	6–10	LLM evalua- tion	NLP	Occasionally
P3	Woman	18–24	U.S.	2–5	LLMs + soci- ety	NLP	Never
P4	Man	25–34	Canada	2–5	Multilingual NLP	NLP	Occasionally
P5	Woman	25–34	U.S.	6–10	Human-AI in- teraction	HCI	Occasionally
P6	Man	25–34	U.S.	2–5	Multimodal AI & HCI	HCI	Occasionally
P7	Woman	25–34	South Korea	2–5	LLM retrieval	NLP	Occasionally
P8	Man	25–34	U.S.	6–10	Human-AI in- teraction	NLP	Occasionally
P9	Woman	18–24	U.S.	0–1	Creativity support tools	HCI	A couple times

Table C.1: Participants from our formative study. All participants were Ph.D. students. **Country** refers to the country in which the participant primarily conducts research at the time of the study. **Research YoE** refers to the years of experience conducting academic research. **AI Use Frequency** refers to how frequently the participants use AI tools to ideate and/or iterate on research ideas. Participants selected “Occasionally” based on the description “I don’t rely on AI but sometimes tinker with it.”

self-defined plan, with the option of invoking the help of an AI assistant through requests prefaced with a “!” command. The study facilitator (the Wizard) entered this command

into Perplexity AI and pasted the output to the other Doc, the “AI output document.” While waiting for the response, we asked participants to write a brief plan consisting of 3–6 steps for how they would complete the request themselves. Participants then inspected the assistant’s output and incorporated any useful text into their document. This process repeated until the participant felt like their idea was sufficiently concrete to write a short descriptive paragraph about it, typically after at least 2–3 requests to the AI. This activity helped us answer **F3**.

C.2.3 Concluding interview

The study ended with a concluding interview, where we asked participants about their experiences, perceptions, and desiderata for the AI assistant, along with preferred ways to interact with it. Each participant was given a \$35 USD honorarium after the study. The study was reviewed and approved by the Allen Institute for AI’s IRB.

C.3 Formative Study Data Analysis

To answer **F1**, the first author used a hybrid inductive-deductive coding process [188] to code participants’ submitted project documents. The first 5 documents were inductively coded to surface common structural elements before the elements were deductively applied to the remaining 4. We iterated on existing elements and added new ones as needed.

For **F2**, we performed inductive thematic analysis on two documents—participants’ submitted project documents and in-study planning documents. The first author sourced *intentions for planning* within submitted project documents by locating the snippet of text that initiates a plan. For example, if the plan consists of a bulleted list, the intention is often expressed in the lead-in text that immediately precedes the list. We note that an individual plan item can also signal planning intent if it contains a nested plan. The first author performed thematic analysis by inductively coding the extracted text. The first author then identified and extracted *plan steps* participants wrote in both documents before inductively coding them.

For **F3**, the first author performed open coding on the study transcripts before thematically analyzing the coded snippets. To enrich the data, the first author also extracted and

inductively coded all requests to the assistant from participants’ planning documents. The codes were discussed and iterated upon with other project team members on a weekly basis over the course of 3 weeks.

C.4 Further Implementation Details

C.4.1 Underlying LLM Agent

Research support tools

We created a custom tool-calling LLM agent¹ powered by GPT-4o. While GPT-4o provides APIs for managing multi-turn conversation context, the system manages its own context using a JSON schema we developed for storing key insights from papers and other forms of output without storing the full artifact. Previous plan steps and their outputs are also represented in the schema to enrich context. To provide high-quality research support, we incorporated two best-in-class research support functionalities into our agent. The first tool allowed the agent to access The Semantic Scholar Open Data Platform [328] via its publicly available search APIs.² This tool provided the agent access to the full text and metadata (such as authors and publication venues) of millions of research papers, created by a state-of-the-art PDF extraction and knowledge graph normalization pipeline, allowing the agent to search for authors and papers based on topics, titles, and rich metadata [328]. The second tool is the *Ask this Paper*³ feature on Semantic Scholar which generates paper summaries based on the paper’s full text given an aspect of interest. This tool allowed the agent to summarize a relevant paper based on specific aspects of interest within a plan, and also synthesize specific aspects across many papers. For example, the agent can use the first tool to find papers relevant to a topic of interest from a particular author, then use the second tool to summarize the findings of each paper, and, finally, synthesize the individual summaries into a compare and contrast statement about the set of papers for the user to review. All prompts used are provided in our Supplementary Materials.

¹<https://platform.openai.com/docs/guides/function-calling>

²<https://www.semanticscholar.org/product/api>

³<https://www.semanticscholar.org/product>

C.4.2 Guidance for plan generation

We use findings from our formative study to guide the agent’s generation of initial plans. In our formative study, we had asked participants to write down brief plans when interacting with our probe (see Appendix C.2). Based on participants’ preferences of which steps they prefer assigning to an AI vs. keeping for themselves (**F3** from our formative study), we organized these steps into agent and user steps. We then wrote examples of how these steps are composed into plans for tackling particular questions, once again drawing from plans participants wrote in the probe activity. These examples were provided for in-context learning in COCOA’s system prompt for plan generation (see prompts in our Supplementary Materials). Finally, we also enable COCOA to learn from interactive plans the user has previously created and edited. When the agent is invoked, COCOA collects the existing plans in the document and adds them to the in-context learning examples on-the-fly. This allows the user’s co-planning efforts to be reused for similar requests elsewhere in the document.

C.5 Lab Study Participants

See Table C.2.

C.6 Lab Study Recruitment Process and Pilot Studies

We sent out a recruitment form to collect basic demographic details, frequency of AI use in research, and a copy of a project document from an ongoing or completed project to use during the study. One participant also participated in our formative study. 15 researchers were based in the United States, and 1 was based in Canada. Participants’ research areas ranged from large language models, to ubiquitous computing, to visualization; broadly, they spanned HCI ($n = 10$), ML ($n = 3$), and NLP ($n = 3$). In our participant pool, many users were occasional (6) or frequent⁴ (6) users of AI tools in the research process.

We also recruited 4 additional participants (1 female, 2 male, 1 non-binary) from our institution for pilot studies to test out COCOA, catch usability issues, and help us refine our procedure. We learned from our pilot studies that the length of generated plans should be

⁴We define “occasional” and “frequent” the same way as we do in our formative study (Table C.1).

P#	Gender	Age Range	Country	Job Title	Res. YoE	Research Area	Area (Gen.)	AI Use Freq.
P1	Woman	25–34	U.S.	Ph.D. student	2–5	Social computing	HCI	Frequently
P2	Woman	25–34	U.S.	Ph.D. student	2–5	Social computing	HCI	Occasionally
P3	Woman	25–34	U.S.	Ph.D. student	6–10	Visualization	HCI	Occasionally
P4	Woman	25–34	U.S.	Ph.D. student	2–5	Culture + com- puting	HCI	A couple times
P5	Woman	25–34	U.S.	Postdoc	6–10	Software develop- ment	HCI	Occasionally
P6	Woman	25–34	U.S.	Ph.D. student	6–10	Health + com- puting	HCI	Occasionally
P7	Man	25–34	U.S.	Ph.D. student	2–5	LLMs	NLP	Frequently
P8	Woman	25–34	U.S.	Ph.D. student	2–5	Accessibility	HCI	Never
P9	Man	25–34	U.S.	Ph.D. student	2–5	Multimodal AI	ML	Frequently
P10	Woman	18–24	U.S.	Ph.D. student	2–5	On-device AI	ML	A couple times
P11	Man	18–24	U.S.	Ph.D. student	2–5	Ubiquitous com- puting	HCI	Always
P12	Man	25–34	U.S.	Ph.D. student	6–10	LLM evaluation	NLP	Frequently
P13	Man	35–44	Canada	Ph.D. student	2–5	Computational biology	ML	Frequently
P14	Woman	18–24	U.S.	Ph.D. student	2–5	LLMs	NLP	Frequently
P15	Woman	25–34	U.S.	Ph.D. student	2–5	Social computing	HCI	Occasionally
P16	Man	25–34	U.S.	Postdoc	6–10	Human-AI inter- action	HCI	Occasionally

Table C.2: Participants from our user study. **Country** refers to the country in which the participant primarily conducts research at the time of the study. **Res. YoE** refers to the years of experience conducting academic research. **AI Use Freq.** refers to how frequently the participants use AI tools to ideate and/or iterate on research ideas. Participants selected “Occasionally” based on the description “I don’t rely on AI but sometimes tinker with it.”

around 3 steps (rather than the system’s default of 5–6 steps) to fit within the study’s time constraints.

C.7 Lab Study Procedural Details

To control for the length of the study, the first author reviewed the participants’ documents prior to the study to identify and highlight two candidate tasks. To make sure the two tasks are valid and comparable, early on in the study, the first author asked participants 1) whether the tasks have been decisively solved by the participant and whether they would still like to explore them in the study, and 2) whether the two tasks are similar in scope, such that one task does not take significantly more resources and effort to explore than the other. If these conditions were not met, the participant was prompted to rewrite one or both tasks until they were. The two tasks were then randomly assigned to COCOA and the baseline.

In the COCOA condition, participants first watched a 6-minute video demo of the system, and were then directed to a document where they would get used to using COCOA’s features on the following sample question: *“How can we use AI to better society?”* This practice period lasted around 10 minutes. Participants were then asked to spend 25 minutes to make as much progress as possible tackling the task assigned to this condition with COCOA, thinking aloud as they did so. They were also permitted to use other tools (academic search engines, other AI tools, social media, etc.) alongside COCOA, although very few did. After the 25-minute session, participants wrote a brief set of takeaways and next steps from their exploration, and filled out a short evaluation form with 5-point Likert scale questions about their experience (see Appendix C.8 for this form). After submitting the form, they were encouraged to further try out COCOA on other parts of their project document for another 10–15 minutes.

In the baseline condition, we did not provide participants a tutorial because the chat interface was already ubiquitous beyond interacting with AI agents. Participants were once again asked to spend 25 minutes making as much progress as possible tackling the task assigned to this condition, thinking aloud as they did so and using other tools if desired. They wrote brief takeaways and next steps after the 25 minutes was up and filled out the

same evaluation form as the COCOA condition.

Because participants identified literature search and understanding as one of the areas they can benefit most from AI assistance (Section 6.3.2), we had participants tackle a literature-augmented task in both COCOA and the baseline. By *literature-augmented*, we mean tasks that are to be completed by referencing or drawing from academic literature; in the study, participants worked on not only literature review tasks, but also tasks where researchers needed to make literature-informed decisions such as study design and ideation. This stands in contrast with tasks focused on writing mechanics (e.g., rewording a sentence) or tasks that do not involve literature (e.g., booking travel) that past work has covered (e.g., [320, 369, 409, 698, 705, 727]).

C.8 Lab Study Self-Evaluation Form Questions

All questions were answered on a 5-point Likert scale.

- The system’s outputs were useful to me for exploring the research problem at hand
- I obtained new and useful insights from using the system
- The system helped me clearly understand the steps needed to effectively tackle this particular problem
- By using the system, I’ve developed a better understanding of potential solutions to this particular problem
- I found the system easy to use
- I could easily steer the system towards doing something helpful
- I could see myself easily integrating this system into my workflow
- I’m satisfied with the artifact (summary and next steps) that I’ve created after using the system
- I feel confident about my next steps after using the system

C.9 Deployment Study General Guidelines

1. Try to spend a total of at least 90 minutes with Cocoa during the 7-day use period.
2. It may help to jot down any notable experiences with Cocoa so we can more easily discuss them later.
3. Treat the document as a scratch space to tinker with and explore rough ideas.
4. Cocoa is just another tool in your toolbox. You can use it alongside whatever you typically use in your workflows!
5. If the system crashes at any point, try refreshing the page. The document auto-saves regularly.

BIBLIOGRAPHY

- [1] Z. Abrams. Artificial intelligence is impacting the field. *Monitor on Psychology*, 56(1):46, 2025.

- [2] OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Ryan Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Adeola Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, Oleg Murk, David M'ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O'Keefe, Jakub W. Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres,

Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorný, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario D. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin D. Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas A. Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer'on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll L. Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lillian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report. 2023.

- [3] Mark S Ackerman. The intellectual challenge of cscw: the gap between social requirements and technical feasibility. *Human-Computer Interaction*, 15(2-3):179–203, 2000.
- [4] Lama Ahmad, Sandhini Agarwal, Michael Lampe, and Pamela Mishkin. Openai’s approach to external red teaming for ai models and systems. *arXiv preprint arXiv:2503.16431*, 2025.
- [5] AI Objectives Institute. Introducing Talk to the City: Collective Deliberation at Scale. <https://ai.objectives.institute/blog/introducing-talk-to-the-city-our-collective-deliberation-tool>, 2023.
- [6] Leah Hope Ajmani, Nuredin Ali Abdelkadir, and Stevie Chancellor. Secondary stakeholders in ai: Fighting for, brokering, and navigating agency. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pages 1095–1107, 2025.
- [7] Robert St. Amant. Navigation and planning in a mixed-initiative user interface. In *AAAI/IAAI*, 1997.
- [8] Vamshi Ambati, Stephan Vogel, and Jaime G. Carbonell. Collaborative workflow for crowdsourcing translation. *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, 2012.

- [9] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 337–346, 2015.
- [10] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. Guidelines for human-ai interaction. CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery.
- [11] Saleema Amershi, Daniel S. Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi T. Iqbal, Paul N. Bennett, Kori Inkpen Quinn, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. Guidelines for human-ai interaction. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.
- [12] Markus Anderljung, Joslyn Barnhart, Anton Korinek, Jade Leung, Cullen O’Keefe, Jess Whittlestone, Shahar Avin, Miles Brundage, Justin Bullock, Duncan Cass-Beggs, et al. Frontier ai regulation: Managing emerging risks to public safety. *arXiv preprint arXiv:2307.03718*, 2023.
- [13] Ida-Elisabeth Andersen and Birgit Jæger. Scenario workshops and consensus conferences: towards more democratic decision-making. *Science and public policy*, 26(5):331–340, 1999.
- [14] Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah D Goodman. Star-gate: Teaching language models to ask clarifying questions. *arXiv preprint arXiv:2403.19154*, 2024.
- [15] Tyler Angert, Miroslav Suzara, Jenny Han, Christopher Pondoc, and Hariharan Subramonyam. Spellburst: A node-based interface for exploratory creative coding with natural language prompts. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023.
- [16] Tyler Angert, Miroslav Suzara, Jenny Han, Christopher Pondoc, and Hariharan Subramonyam. Spellburst: A node-based interface for exploratory creative coding with natural language prompts. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST ’23, New York, NY, USA, 2023. Association for Computing Machinery.
- [17] Jacy Reese Anthis, Ryan Liu, Sean M Richardson, Austin C Kozlowski, Bernard Koch, James Evans, Erik Brynjolfsson, and Michael Bernstein. Llm social simulations are a promising research method. *arXiv preprint arXiv:2504.02234*, 2025.

- [18] Anthropic. Claude’s Constitution. <https://www.anthropic.com/news/claudes-constitution>, 2023.
- [19] Anthropic. Claude’s Constitution. <https://www.anthropic.com/news/claudes-constitution>, 03 2023.
- [20] Anthropic. Collective Constitutional AI: Aligning a Language Model with Public Input. <https://www.anthropic.com/news/collective-constitutional-ai-aligning-a-language-model-with-public-input>, 2023.
- [21] Anthropic. Building effective agents, 2024.
- [22] Anthropic. Developing a computer use model. <https://www.anthropic.com/news/developing-computer-use>, 2024.
- [23] Anthropic. Responsible Scaling Policy. <https://assets.anthropic.com/m/24a47b00f10301cd/original/Anthropic-Responsible-Scaling-Policy-2024-10-15.pdf>, 2024.
- [24] Anthropic. Giving Claude a role with a system prompt. <https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/system-prompts>, 2025.
- [25] Anthropic. System Card: Claude Opus 4 & Claude Sonnet 4. <https://www-cdn.anthropic.com/6be99a52cb68eb70eb9572b4cafad13df32ed995.pdf>, 05 2025.
- [26] Anthropic. Use XML tags to structure your prompts. <https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/use-xml-tags>, 2025.
- [27] Anthropic. Claude’s Constitution. https://www-cdn.anthropic.com/d0636f72a9493d279ed36b33987da3430bcb5911/claudes-constitution_webPDF_26-02.02a.pdf, 2026.
- [28] Anthropic Frontier Red Team. Patterns and problems in emerging multiagent systems. <https://www.anthropic.com/research/multiagent-systems>, 2026.
- [29] Apple. Machine Learning—Human Interface Guidelines. <https://developer.apple.com/design/human-interface-guidelines/technologies/machine-learning/introduction>, 2023.
- [30] Ian Arawjo, Chelse Swoopes, Priyan Vaithilingam, Martin Wattenberg, and Elena L. Glassman. Chainforge: A visual toolkit for prompt engineering and llm hypothesis testing. *ArXiv*, abs/2309.09128, 2023.

- [31] Ian Arawjo, Chelse Swoopes, Priyan Vaithilingam, Martin Wattenberg, and Elena L Glassman. Chainforge: A visual toolkit for prompt engineering and llm hypothesis testing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2024.
- [32] Ian Arawjo, Chelse Swoopes, Priyan Vaithilingam, Martin Wattenberg, and Elena L Glassman. Chainforge: A visual toolkit for prompt engineering and llm hypothesis testing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2024.
- [33] Sveinung Arnesen, Troy Saghaug Broderstad, James Fishkin, Mikael Poul Johanneson, and Alice Siu. Knowledge and support for ai in the public sector: a deliberative poll experiment. *AI Soc.*, 40:3573–3589, 2024.
- [34] Artificial Analysis. Independent analysis of AI. <https://artificialanalysis.ai/>, 2026.
- [35] Isaac Asimov. *I. robot*. Narkaling Productions., 1940.
- [36] AutoGPT. Empower your digital tasks with AutoGPT. <https://agpt.co/>, 2024.
- [37] Amid Ayobi, Jacob Hughes, Christopher Duckworth, Jakub J Dylag, Sam James, Paul Marshall, Matthew Guy, Anitha Kumaran, Adriane Chapman, Michael J. Boniface, and Aisling Ann O’Kane. Computational notebooks as co-design tools: Engaging young adults living with diabetes, family carers, and clinicians with machine learning models. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.
- [38] Tamara Babaian, Barbara J. Grosz, and Stuart M. Shieber. A writer’s collaborative assistant. In *International Conference on Intelligent User Interfaces*, 2002.
- [39] Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. *ArXiv*, abs/2404.07738, 2024.
- [40] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova Das-Sarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- [41] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

- [42] Jonathan Ball. The double diamond: A universally accepted depiction of the design process. <https://www.designcouncil.org.uk/news-opinion/double-diamond-universally-accepted-depiction-design-process>, 2005.
- [43] Jonathan Ball. The double diamond: A universally accepted depiction of the design process. <https://www.designcouncil.org.uk/news-opinion/double-diamond-universally-accepted-depiction-design-process>, 2005.
- [44] Gagan Bansal, Jennifer Wortman Vaughan, Saleema Amershi, Eric Horvitz, Adam Fourney, Hussein Mozannar, Victor Dibia, and Daniel S Weld. Challenges in human-agent communication. 2024.
- [45] Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–16, 2021.
- [46] Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.
- [47] Jakob E Bardram and Claus Bossen. A web of coordinative artifacts: collaborative work at a hospital ward. In *Proceedings of the 2005 ACM International Conference on Supporting Group Work*, pages 168–176, 2005.
- [48] Julia Barnett, Kimon Kieslich, Natali Helberger, and Nicholas Diakopoulos. Envisioning stakeholder-action pairs to mitigate negative impacts of ai: A participatory approach to inform policy making. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pages 1424–1449, 2025.
- [49] Julia Barnett, Kimon Kieslich, Jasmine Sinchai, and Nicholas Diakopoulos. Scenarios in computing research: A systematic review of the use of scenario methods for exploring the future of computing technologies in society. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(1):316–329, Oct. 2025.
- [50] Alex Bäuerle, Ángel Alexander Cabrera, Fred Hohman, Megan Maher, David Koski, Xavier Suau, Titus Barik, and Dominik Moritz. Symphony: Composing interactive interfaces for machine learning. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.
- [51] Michel Beaudouin-Lafon and Wendy Mackay. Prototyping tools and techniques. In *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, pages 1006–1031. 2002.

- [52] George A Bekey. *Autonomous robots: from biological inspiration to implementation and control*. MIT press, 2005.
- [53] Yoshua Bengio, Sören Mindermann, Daniel Privitera, Tamay Besiroglu, Rishi Bommasani, Stephen Casper, Yejin Choi, Philip Fox, Ben Garfinkel, Danielle Goldfarb, et al. International ai safety report. *arXiv preprint arXiv:2501.17805*, 2025.
- [54] Jesse Josua Benjamin, Arne Berger, Nick Merrill, and James Pierce. Machine learning uncertainty as a design material: A post-phenomenological inquiry. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.
- [55] Dan Bennett, Oussama Metatla, Anne Roudaut, and Elisa D Mekler. How does hci understand human agency and autonomy? In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2023.
- [56] Dan Bennett, Oussama Metatla, Anne Roudaut, and Elisa D. Mekler. How does hci understand human agency and autonomy? *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.
- [57] Philipp Berens, Kyle Cranmer, Neil D. Lawrence, Ulrike von Luxburg, and Jessica Montgomery. Ai for science: An emerging agenda. *ArXiv*, abs/2303.04217, 2023.
- [58] Michael S. Bernstein, Greg Little, Rob Miller, Bjoern Hartmann, Mark S. Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, 2010.
- [59] Hugh Beyer and Karen Holtzblatt. Contextual design. *interactions*, 6(1):32–42, 1999.
- [60] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Rob Miller, Robin Miller, Aubrey Tatarowicz, Brandyn Allen White, Samuel White, and Tom Yeh. Vizwiz: nearly real-time answers to visual questions. *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, 2010.
- [61] Abeba Birhane, William Isaac, Vinodkumar Prabhakaran, Mark Diaz, Madeleine Clare Elish, Iason Gabriel, and Shakir Mohamed. Power to the people? opportunities and challenges for participatory ai. In *Proceedings of the 2nd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–8, 2022.
- [62] Anna CS Bodén, Jesper Molin, Stina Garvin, Rebecca A West, Claes Lundström, and Darren Treanor. The human-in-the-loop: an evaluation of pathologists’ interaction with artificial intelligence in clinical practice. *Histopathology*, 79(2):210–218, 2021.

- [63] Susanne Bodker. Scenarios in user-centred design-setting the stage for reflection and action. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers*, pages 11–pp. IEEE, 1999.
- [64] R. J. Bogumil. The reflective practitioner: How professionals think in action. *Proceedings of the IEEE*, 73:845–846, 1985.
- [65] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- [66] Rafael H Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.
- [67] Sam Bowman. The Checklist: What Succeeding at AI Safety Will Involve. <https://sleepinyourhat.github.io/checklist/>, 2024.
- [68] Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiuotė, Amanda Askell, Andy Jones, Anna Chen, et al. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022.
- [69] Jonathan Bragg, Mausam, and Daniel S. Weld. Crowdsourcing multi-label classification for taxonomy creation. In *AAAI Conference on Human Computation & Crowdsourcing*, 2013.
- [70] Jonathan Bragg, Mausam, and Daniel S. Weld. Sprout: Crowd-powered task design for crowdsourcing. *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, 2018.
- [71] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV 11*, pages 438–451. Springer, 2010.
- [72] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.
- [73] Virginia Braun and Victoria Clarke. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health*, 11(4):589–597, 2019.

- [74] Virginia Braun and Victoria Clarke. Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, 11:589 – 597, 2019.
- [75] John Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.
- [76] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
- [77] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.
- [78] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [79] Jonah Brown-Cohen, Geoffrey Irving, and Georgios Piliouras. Scalable ai safety via doubly-efficient debate. *arXiv preprint arXiv:2311.14125*, 2023.
- [80] Adrian Bua and Oliver Escobar. Participatory-deliberative processes and public policy agendas: lessons for policy and practice. *Policy Design and Practice*, 1(2):126–140, 2018.
- [81] Marion Buchenau and Jane Fulton Suri. Experience prototyping. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 424–433, 2000.
- [82] Marie Davidsen Buhl, Gaurav Sett, Leonie Koessler, Jonas Schuett, and Markus Anderljung. Safety cases for frontier ai. *arXiv preprint arXiv:2410.21572*, 2024.
- [83] Mark H. Burstein and Drew McDermott. Issues in the development of human-computer mixed initiative planning. 1994.
- [84] Ángel Alexander Cabrera, Erica Fu, Donald Bertucci, Kenneth Holstein, Ameet Talwalkar, Jason I Hong, and Adam Perer. Zeno: An interactive framework for behavioral evaluation of machine learning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2023.
- [85] Alice Cai, Ian Arawjo, and Elena L Glassman. Antagonistic ai. *arXiv preprint arXiv:2402.07350*, 2024.

- [86] Bradley Camburn, Vimal Viswanathan, Julie Linsey, David Anderson, Daniel Jensen, Richard Crawford, Kevin Otto, and Kristin Wood. Design prototyping methods: state of the art in strategies, techniques, and guidelines. *Design Science*, 3:e13, 2017.
- [87] Tara Capel and Margot Brereton. What is human-centered about human-centered ai? a map of the research landscape. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–23, 2023.
- [88] Nicholas A Caputo. Alignment as jurisprudence. *Yale Journal of Law and Technology (forthcoming)*, 2024.
- [89] Stuart K Card, Thomas P Moran, and Allen Newell. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410, 1980.
- [90] Michelle Carney, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. Teachable machine: Approachable web-based tool for exploring machine learning classification. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–8, New York, NY, USA, 2020. Association for Computing Machinery.
- [91] Charles E Carpenter. Court decisions and the common law. *Columbia Law Review*, 17(7):593–607, 1917.
- [92] John M Carrol. Five reasons for scenario-based design. In *Proceedings of the 32nd annual hawaii international conference on systems sciences. 1999. hicss-32. abstracts and cd-rom of full papers*, pages 11–pp. IEEE, 1999.
- [93] John M Carroll. *Making use: scenario-based design of human-computer interactions*. MIT press, 2003.
- [94] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.
- [95] Stephen Casper, Jason Lin, Joe Kwon, Gatlen Culp, and Dylan Hadfield-Menell. Explore, establish, exploit: Red teaming language models from scratch, 2023.
- [96] Louis Castricato, Nathan Lile, Rafael Rafailov, Jan-Philipp Fränken, and Chelsea Finn. Persona: A reproducible testbed for pluralistic alignment. *arXiv preprint arXiv:2407.17387*, 2024.
- [97] Chengliang Chai and Guoliang Li. Human-in-the-loop techniques in machine learning. *IEEE Data Eng. Bull.*, 43(3):37–52, 2020.

- [98] Alan Chan, Carson Ezell, Max Kaufmann, Kevin Wei, Lewis Hammond, Herbie Bradley, Emma Blumke, Nitarshan Rajkumar, David Krueger, Noam Kolt, et al. Visibility into ai agents. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 958–973, 2024.
- [99] Alan Chan, Rebecca Salganik, Alva Markelius, Chris Pang, Nitarshan Rajkumar, Dmitrii Krashennnikov, Lauro Langosco, Zhonghao He, Yawen Duan, Micah Carroll, et al. Harms from increasingly agentic algorithmic systems. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 651–666, 2023.
- [100] Alan Chan, Kevin Wei, Sihao Huang, Nitarshan Rajkumar, Elija Perrier, Seth Lazar, Gillian K Hadfield, and Markus Anderljung. Infrastructure for ai agents. *arXiv preprint arXiv:2501.10114*, 2025.
- [101] Joel Chan, Joseph Chee Chang, Tom Hope, Dafna Shahaf, and Aniket Kittur. Solvent: A mixed initiative system for finding analogies between research papers. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–21, 2018.
- [102] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017.
- [103] Joseph Chee Chang, Amy X. Zhang, Jonathan Bragg, Andrew Head, Kyle Lo, Doug Downey, and Daniel S. Weld. Citesee: Augmenting citations in scientific papers with persistent and personalized historical context. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.
- [104] Crystal Chao, Maya Cakmak, and Andrea L Thomaz. Transparent active learning for robots. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 317–324. IEEE, 2010.
- [105] Souti Chattopadhyay, I. V. R. K. V. Prasad, Austin Z. Henley, Anita Sarma, and Titus Barik. What’s wrong with computational notebooks? pain points, needs, and design opportunities. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.
- [106] Khaoula Chehbouni, Mohammed Haddou, Jackie Chi Kit Cheung, and Golnoosh Farnadi. Neither valid nor reliable? investigating the use of llms as judges. *arXiv preprint arXiv:2508.18076*, 2025.
- [107] Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. Fireact: Toward language agent fine-tuning. *ArXiv*, abs/2310.05915, 2023.

- [108] Jiangjie Chen, Siyu Yuan, Rong Ye, Bodhisattwa Prasad Majumder, and Kyle Richardson. Put your money where your mouth is: Evaluating strategic planning and execution of llm agents in an auction arena. *arXiv preprint arXiv:2310.05746*, 2023.
- [109] Quan Ze Chen, KJ Feng, Chan Young Park, and Amy X Zhang. Spica: Retrieving scenarios for pluralistic in-context alignment. *arXiv preprint arXiv:2411.10912*, 2024.
- [110] Quan Ze Chen, Tobias Schnabel, Besmira Nushi, and Saleema Amershi. Hint: Integration testing for ai-based features with humans in the loop. In *27th International Conference on Intelligent User Interfaces, IUI '22*, page 549–565, New York, NY, USA, 2022. Association for Computing Machinery.
- [111] Quan Ze Chen, Daniel S. Weld, and Amy X. Zhang. Goldilocks: Consistent crowd-sourced scalar annotations with relative uncertainty. *Proceedings of the ACM on Human-Computer Interaction*, 5:1 – 25, 2021.
- [112] Quan Ze Chen and Amy Xian Zhang. Case law grounding: Using precedents to align decision-making for humans and ai. In *Proceedings of the ACM Collective Intelligence Conference, CI '25*, page 226–238, New York, NY, USA, 2025. Association for Computing Machinery.
- [113] Quanze Chen, Jonathan Bragg, Lydia B Chilton, and Dan S Weld. Cicero: Multi-turn, contextual argumentation for accurate crowdsourcing. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–14, 2019.
- [114] Justin Cheng and Michael S. Bernstein. Flock: Hybrid crowd-machine learning classifiers. *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 2015.
- [115] Myra Cheng, Esin Durmus, and Dan Jurafsky. Marked personas: Using natural language prompts to measure stereotypes in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1504–1532, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [116] Ruijia Cheng, Titus Barik, Alan Leung, Fred Hohman, and Jeffrey Nichols. Biscuit: Scaffolding llm-generated code with ephemeral uis in computational notebooks. *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 13–23, 2024.
- [117] Inyoung Cheong, King Xia, K. J. Kevin Feng, Quan Ze Chen, and Amy X. Zhang. (a)i am not a lawyer, but...: Engaging legal experts towards responsible llm policies for legal advice. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability,*

- and Transparency*, FAccT '24, page 2454–2469, New York, NY, USA, 2024. Association for Computing Machinery.
- [118] Inyoung Cheong, King Xia, K. J. Kevin Feng, Quan Ze Chen, and Amy X. Zhang. (a)i am not a lawyer, but...: Engaging legal experts towards responsible llm policies for legal advice. In *Conference on Fairness, Accountability and Transparency*, 2024.
 - [119] Lydia B. Chilton, Greg Little, Darren Edge, Daniel S. Weld, and James A. Landay. Cascade: crowdsourcing taxonomy creation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013.
 - [120] Frederick Choi, Sajjadur Rahman, Han Jun Kim, and Daz Zhang. Towards transparent, reusable, and customizable data science in computational notebooks. *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.
 - [121] Andrew R. Chow. How ChatGPT Managed to Grow Faster Than TikTok or Instagram. <https://time.com/6253615/chatgpt-fastest-growing/>, 2023.
 - [122] Brian Christian. *The alignment problem: Machine learning and human values*. WW Norton & Company, 2020.
 - [123] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
 - [124] Peter Cihon, Merlin Stein, Gagan Bansal, Sam Manning, and Kevin Xu. Measuring ai agent autonomy: Towards a scalable approach with code inspection. *arXiv preprint arXiv:2502.15212*, 2025.
 - [125] Herbert H Clark and Susan E Brennan. Grounding in communication. 1991.
 - [126] Nick Clegg. Bringing People Together to Inform Decision-Making on Generative AI. <https://www.peoplepowered.org/participatory-policymaking>, 06 2023.
 - [127] Nick Clegg. Bringing People Together to Inform Decision-Making on Generative AI. <https://about.fb.com/news/2023/06/generative-ai-community-forum/>, 2023.
 - [128] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. Specter: Document-level representation learning using citation-informed transformers. *ArXiv*, abs/2004.07180, 2020.
 - [129] Katherine M Collins, Ilia Sucholutsky, Umang Bhatt, Kartik Chandra, Lionel Wong, Mina Lee, Cedegao E Zhang, Tan Zhi-Xuan, Mark Ho, Vikash Mansinghka, et al. Building machines that learn and think with people. *arXiv preprint arXiv:2408.03943*, 2024.

- [130] Eric Corbett, Emily Denton, and Sheena Erete. Power and public participation in ai. In *Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–13, 2023.
- [131] Marco Costanzo, Giuseppe De Maria, and Ciro Natale. Handover control for human-robot and robot-robot collaboration. *Frontiers in Robotics and AI*, 8:672995, 2021.
- [132] Anamaria Crisan and Brittany Fiore-Gartland. Fits and starts: Enterprise use of automl and the role of humans in the loop. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.
- [133] Roger Crisp. *Aristotle: nicomachean ethics*. Cambridge University Press, 2014.
- [134] Nigel Cross. Designerly ways of knowing. *Design studies*, 3(4):221–227, 1982.
- [135] Cyrus Moulton. What is Jevons Paradox? And why it may — or may not — predict AI’s future. <https://news.northeastern.edu/2025/02/07/jevons-paradox-ai-future/>, 2025.
- [136] Allan Dafoe, Anca Dragan, Four Flynn, Helen King, Tom Lue, Lewis Ho, and Rohin Shah. Updating the frontier safety framework, February 2025.
- [137] Databricks. What is an AI Agent Harness? <https://www.databricks.com/blog/ai-harness>, 2026.
- [138] Dave Citron. Try Deep Research and our new experimental model in Gemini, your AI assistant. <https://blog.google/products/gemini/google-gemini-deep-research/>, 2024.
- [139] Munmun De Choudhury, Sachin R Pendse, and Neha Kumar. Benefits and harms of large language models in digital mental health. *arXiv preprint arXiv:2311.14693*, 2023.
- [140] Douglas L. Dean, Jillian M. Hender, Thomas Lee Rodgers, and Eric L. Santanen. Identifying quality, novel, and creative ideas: Constructs and scales for idea evaluation. *J. Assoc. Inf. Syst.*, 7:30, 2006.
- [141] Fernando Delgado, Stephen Yang, Michael Madaio, and Qian Yang. The participatory turn in ai design: Theoretical foundations and the current state of practice. In *Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–23, 2023.

- [142] Democracy Beyond Elections. Policy-making needs a reboot. <https://www.democracybeyondelections.org/policy/>, 2021.
- [143] Wesley Hanwen Deng, Wang Claire, Howard Ziyu Han, Jason I Hong, Kenneth Holstein, and Motahhare Eslami. Weaudit: Scaffolding user auditors and ai practitioners in auditing generative ai. *Proceedings of the ACM on Human-Computer Interaction*, 9(7):1–35, 2025.
- [144] Wesley Hanwen Deng, Sunnie SY Kim, Akshita Jha, Ken Holstein, Motahhare Eslami, Lauren Wilcox, and Leon A Gatys. Personateaming: Exploring how introducing personas can improve automated ai red-teaming. *arXiv preprint arXiv:2509.03728*, 2025.
- [145] Wesley Hanwen Deng, Nur Yildirim, Monica Chang, Motahhare Eslami, Kenneth Holstein, and Michael Madaio. Investigating practices and opportunities for cross-functional collaboration around ai fairness in industry practice. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’23*, page 705–716, New York, NY, USA, 2023. Association for Computing Machinery.
- [146] Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. Toxicity in chatgpt: Analyzing persona-assigned language models. *arXiv preprint arXiv:2304.05335*, 2023.
- [147] Alicia DeVrio, Aditi Dhabalia, Hong Shen, Kenneth Holstein, and Motahhare Eslami. Toward user-driven algorithm auditing: Investigating users’ strategies for uncovering harmful algorithmic behavior. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–19, 2022.
- [148] Kaustubh Dhole. Large language models as sociotechnical systems. In *Proceedings of the Big Picture Workshop*, pages 66–79, 2023.
- [149] María Fernanda Díaz and Julian “Iñaki” Goñi. What expertise, for what, and whose democratic politics? *The Oxford handbook of expertise and democratic politics*, edited by Gil Eyal and Thomas Medvetz, 2024.
- [150] Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4537–4546, Hong Kong, China, November 2019. Association for Computational Linguistics.

- [151] Kimberly Do, Rock Yuren Pang, Jiachen Jiang, and Katharina Reinecke. “that’s important, but...”: How computer science researchers anticipate unintended consequences of their research innovations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI ’23, New York, NY, USA, 2023. Association for Computing Machinery.
- [152] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.
- [153] Paul Dourish and Victoria Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 107–114, 1992.
- [154] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. Ux design innovation: Challenges for working with machine learning as a design material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI ’17, page 278–288, New York, NY, USA, 2017. Association for Computing Machinery.
- [155] Steven P Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L Schwartz, and Scott R Klemmer. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(4):1–24, 2010.
- [156] Steven W Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L. Schwartz, and Scott R. Klemmer. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Trans. Comput. Hum. Interact.*, 17:18:1–18:24, 2010.
- [157] Ruofei Du, Na Li, Jing Jin, Michelle Carney, Scott Miles, Maria Kleiner, Xiuxiu Yuan, Yinda Zhang, Anuva Kulkarni, Xingyu Liu, Ahmed Sabie, Sergio Orts-Escolano, Abhishek Kar, Ping Yu, Ram Iyengar, Adarsh Kowdle, and Alex Olwal. Rapsai: Accelerating machine learning prototyping of multimedia applications through visual programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI ’23, New York, NY, USA, 2023. Association for Computing Machinery.
- [158] Batu El, Shiye Su, Aneesh Pappu, Peggy Yin, Julie Heng, Eric Heng, Ryan Z Wang, Andreas Haupt, and James Zou. Position: Ai development should prioritize cognitive security. In *ICLR 2026 Workshop on Agents in the Wild (AIWILD)*, 2026.
- [159] Elicit. Elicit: The AI Research Assistant. <https://elicit.com/>, 2024.
- [160] Elicit. AI for systematic reviews & meta-analyses. <https://elicit.com/solutions/systematic-reviews>, 2025.

- [161] Elman. Unlock R&D superpowers. <https://elman.ai/>, 2024.
- [162] Chris Elsdén, Ella Tallyn, and Bettina Nissen. When do design workshops work (or not)? In *Companion publication of the 2020 ACM designing interactive systems conference*, pages 245–250, 2020.
- [163] Jon Elster. *Deliberative democracy*, volume 1. Cambridge University Press, 1998.
- [164] Simeon Emanuilov. Devin, the AI Software Engineer: Review. https://unfoldai.com/devin-the-ai-software-engineer-review/#Limitations_and_challenges, 2024.
- [165] Douglas Engelbart. Augmenting human intellect: A conceptual framework. summary report. *Stanford Research Institute, on Contract AF*, 49(638):1024, 1962.
- [166] Thomas Erickson and Wendy A Kellogg. Social translucence: an approach to designing systems that support social processes. *ACM transactions on computer-human interaction (TOCHI)*, 7(1):59–83, 2000.
- [167] Motahhare Eslami, Karrie Karahalios, Christian Sandvig, Kristen Vaccaro, Aimee Rickman, Kevin Hamilton, and Alex Kirlik. First i "like" it, then i hide it: Folk theories of social feeds. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 2371–2382, New York, NY, USA, 2016. Association for Computing Machinery.
- [168] Jerry Alan Fails and Dan R. Olsen. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, page 39–45, New York, NY, USA, 2003. Association for Computing Machinery.
- [169] Jenny Fan and Amy X Zhang. Digital juries: A civics-oriented approach to platform governance. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–14, 2020.
- [170] Richard Fang, Dylan Bowman, and Daniel Kang. Voice-enabled ai agents can perform common scams. *arXiv preprint arXiv:2410.15650*, 2024.
- [171] Mehdi Farashahi and Mahdi Tajeddin. Effectiveness of teaching methods in business education: A comparison study on the learning outcomes of lectures, case studies and simulations. *The international journal of Management Education*, 16(1):131–142, 2018.
- [172] Michael Feffer, Anusha Sinha, Wesley H Deng, Zachary C Lipton, and Hoda Heidari. Red-teaming for generative ai: Silver bullet or security theater? In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 421–437, 2024.

- [173] K. J. Kevin Feng, Quan Ze Chen, Inyoung Cheong, King Xia, and Amy X. Zhang. Case repositories: Towards case-based reasoning for ai alignment. *ArXiv*, abs/2311.10934, 2023.
- [174] K. J. Kevin Feng, Inyoung Cheong, Quan Ze Chen, and Amy X. Zhang. Policy prototyping for llms: Pluralistic alignment via interactive and collaborative policymaking. *arXiv preprint arXiv:2409.08622*, 2024.
- [175] K. J. Kevin Feng, Inyoung Cheong, Quan Ze Chen, and Amy X. Zhang. Policy prototyping for llms: Pluralistic alignment via interactive and collaborative policymaking. *ArXiv*, abs/2409.08622, 2024.
- [176] Kevin Feng, Tae Soo Kim, Rock Yuren Pang, Faria Huq, Tal August, and Amy X Zhang. On the regulatory potential of user interfaces for ai agent governance. In *NeurIPS 2025 Workshop on Regulatable ML*.
- [177] KJ Feng, Quan Ze Chen, Inyoung Cheong, King Xia, and Amy X Zhang. Case repositories: Towards case-based reasoning for ai alignment. *arXiv preprint arXiv:2311.10934*, 2023.
- [178] KJ Feng, Q Vera Liao, Ziang Xiao, Jennifer Wortman Vaughan, Amy X Zhang, and David W McDonald. Canvil: Designerly adaptation for llm-powered user experiences. *arXiv preprint arXiv:2401.09051*, 2024.
- [179] KJ Feng, David W McDonald, and Amy X Zhang. Levels of autonomy for ai agents. *arXiv preprint arXiv:2506.12469*, 2025.
- [180] KJ Feng, Kevin Pu, Matt Latzke, Tal August, Pao Siangliulue, Jonathan Bragg, Daniel S Weld, Amy X Zhang, and Joseph Chee Chang. Cocoa: Co-planning and co-execution with ai agents. *arXiv preprint arXiv:2412.10999*, 2024.
- [181] KJ Kevin Feng, Maxwell James Coppock, and David W McDonald. How do ux practitioners communicate ai as a design material? artifacts, conceptions, and propositions. In *Proceedings of the 2023 ACM designing interactive systems conference*, pages 2263–2280, 2023.
- [182] KJ Kevin Feng, Tony W Li, and Amy X Zhang. Understanding collaborative practices and tools of professional ux practitioners in software organizations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2023.
- [183] KJ Kevin Feng and David W McDonald. Addressing ux practitioners’ challenges in designing ml applications: an interactive machine learning approach. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 337–352, 2023.

- [184] KJ Kevin Feng, Rock Yuren Pang, Tzu-Sheng Kuo, Amy Winecoff, Emily Tseng, David Gray Widder, Harini Suresh, Katharina Reinecke, and Amy X Zhang. Sociotechnical ai governance: Challenges and opportunities for hci. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2025.
- [185] KJ Kevin Feng and Amy X. Zhang. From handoffs to co-creation: Deepening collaboration between designers, developers, and data science workers in ux design. In *Proceedings of the InContext: Futuring User-Experience Design Tools Workshop at CHI Conference on Human Factors in Computing Systems (CHI '22)*, 2022.
- [186] Shangbin Feng, Taylor Sorensen, Yuhan Liu, Jillian R. Fisher, Chan Young Park, Yejin Choi, and Yulia Tsvetkov. Modular pluralism: Pluralistic alignment via multi-llm collaboration. *ArXiv*, abs/2406.15951, 2024.
- [187] Jacques Ferber and Gerhard Weiss. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-wesley Reading, 1999.
- [188] Jennifer Fereday and Eimear Muir-Cochrane. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods*, 5(1):80–92, 2006.
- [189] George Ferguson, James F. Allen, Bradford W. Miller, and Eric K. Ringger. The design and implementation of the trains-96 system: A prototype mixed-initiative planning assistant. 1996.
- [190] Figma. Branching in Figma. <https://www.figma.com/best-practices/branching-in-figma/>, 2024.
- [191] Figma. Guide to the Figma Community. <https://help.figma.com/hc/en-us/articles/360038510693-Guide-to-the-Figma-Community>, 2024.
- [192] Figma. How to make a storyboard for UX design in 5 step. <https://www.figma.com/resource-library/how-to-create-a-ux-storyboard/>, 2025.
- [193] Figma. What is prototyping. <https://www.figma.com/resource-library/what-is-prototyping/>, 2025.
- [194] James S Fishkin and Robert C Luskin. Experimenting with a democratic ideal: Deliberative polling and public opinion. *Acta politica*, 40:284–298, 2005.
- [195] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

- [196] Raymond Fok, Joseph Chee Chang, Tal August, Amy X. Zhang, and Daniel S. Weld. Qlarify: Recursively expandable abstracts for directed information retrieval over scientific papers. 2023.
- [197] Raymond Fok, Hita Kambhamettu, Luca Soldaini, Jonathan Bragg, Kyle Lo, Andrew Head, Marti A. Hearst, and Daniel S. Weld. Scim: Intelligent skimming support for scientific papers. *Proceedings of the 28th International Conference on Intelligent User Interfaces*, 2022.
- [198] Matthew C. Fontaine and Stefanos Nikolaidis. Evaluating human–robot interaction algorithms in shared autonomy via quality diversity scenario generation. *ACM Transactions on Human-Robot Interaction (THRI)*, 11:1 – 30, 2022.
- [199] Allen Institute for AI. Semantic Scholar | AI Powered Research Tool. <https://semanticscholar.org/>, 2024.
- [200] Patrick Fournier. *When citizens decide: Lessons from citizen assemblies on electoral reform*. Oxford University Press, 2011.
- [201] Kai Fronsdal, Isha Gupta, Abhay Sheshadri, Jonathan Michala, Stephen McAleer, Rowan Wang, Sara Price, and Samuel R Bowman. Petri: an open-source auditing tool to accelerate ai safety research. *Anthropic Alignment Science Blog*, 10, 2025.
- [202] Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and machines*, 30(3):411–437, 2020.
- [203] Anders Gammelgård-Larsen, Niels van Berkel, Mikael B. Skov, and Jesper Kjeldskov. Designing for human-ai interaction: Comparing intermittent, continuous, and proactive interactions for a music application. In *CHI Extended Abstracts*, 2024.
- [204] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom Brown, Nicholas Joseph, Sam McCandlish, Chris Olah, Jared Kaplan, and Jack Clark. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned, 2022.
- [205] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

- [206] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online, August 2021. Association for Computational Linguistics.
- [207] Thomas Germain. Google just updated its algorithm. The Internet will never be the same. <https://www.bbc.com/future/article/20240524-how-googles-new-algorithm-will-shape-your-internet>, 2024.
- [208] Elliot Gestrin, Marco Kuhlmann, and Jendrik Seipp. Nl2plan: Robust llm-driven planning from minimal text descriptions. *ArXiv*, abs/2405.04215, 2024.
- [209] Tarleton Gillespie. *Custodians of the Internet: Platforms, content moderation, and the hidden decisions that shape social media*. Yale University Press, 2018.
- [210] GitHub. What are AI agents?, 2025.
- [211] M Glez-Bedia, JM Corchado, ES Corchado, and C Fyfe. Analytical model for constructing deliberative agents. *Engineering Intelligent Systems for Electrical Engineering and Communications*, 10(3):173–185, 2002.
- [212] Jody Godoy. Us senators call for meta probe after reuters report on its ai policies. <https://www.reuters.com/legal/litigation/us-senators-call-meta-probe-after-reuters-report-its-ai-policies-2025-08-14/>, 08 2025.
- [213] Arthur Goemans, Marie Davidsen Buhl, Jonas Schuett, Tomek Korbak, Jessica Wang, Benjamin Hilton, and Geoffrey Irving. Safety case template for frontier ai: A cyber inability argument. *arXiv preprint arXiv:2411.08088*, 2024.
- [214] Beth Goldberg, Diana Acosta-Navas, Michiel Bakker, Ian Beacock, Matt Botvinick, Prateek Buch, Renée DiResta, Nandika Donthi, Nathanael Fast, Ravi Iyer, Zaria Jalan, Andrew Konya, Grace Kwak Danciu, Hélène Landemore, Alice Marwick, Carl Miller, Aviv Ovadya, Emily Saltz, Lisa Schirch, Dalit Shalom, Divya Siddarth, Felix Sieker, Christopher Small, Jonathan Stray, Audrey Tang, Michael Henry Tessler, and Amy Zhang. Ai and the future of digital public squares, 2024.
- [215] Google. People + AI Guidebook. <https://pair.withgoogle.com/guidebook/>, 2021.
- [216] Google. Personalized AI for you. https://www.youtube.com/watch?v=v5tRc_5-8G4, 2023.
- [217] Google. Cloud automl custom machine learning models. <https://cloud.google.com/automl/>, 2024.

- [218] Google. Google Scholar. <https://scholar.google.com/>, 2024.
- [219] Google. Gemini 2.5 ProModel Card. <https://storage.googleapis.com/model-cards/documents/gemini-2.5-pro.pdf>, 06 2025.
- [220] Google. Gemini Deep Research. <https://gemini.google/overview/deep-research/>, 2025.
- [221] Google. Get more done with Gemini in Docs. <https://workspace.google.com/products/docs/ai/>, 2025.
- [222] Jochen Görtler, Fred Hohman, Dominik Moritz, Kanit Wongsuphasawat, Donghao Ren, Rahul Nair, Marc Kirchner, and Kayur Patel. Neo: Generalizing confusion matrix visualization to hierarchical and multi-output labels. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.
- [223] Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.
- [224] Graham Wetherall-Grujić. How Belgium is Giving Citizens a Say on AI. *Democracy Technology*, 2024.
- [225] Judith Gregory. Scandinavian approaches to participatory design. *International Journal of Engineering Education*, 19(1):62–74, 2003.
- [226] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hananeh Hajishirzi. Olmo: Accelerating the science of language models. *Preprint*, 2024.
- [227] Konstantin Grotov, Sergey Titov, Yaroslav Zharov, and Timofey Bryksin. Untangling knots: Leveraging llm for error resolution in computational notebooks. *arXiv preprint arXiv:2405.01559*, 2024.
- [228] Lara Groves, Andrew Strait, Aidan Peppin, Octavia Field Reid, Jenny Brennan, and Kira Allmann. Going public: Exploring public participation in commercial AI labs. <https://www.adalovelaceinstitute.org/report/going-public-participation-ai/>, 2023.

- [229] Jonathan Grudin. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM*, 37(1):92–105, 1994.
- [230] Jonathan Grudin. A moving target: The evolution of human-computer interaction. *Human-computer interaction handbook*, 3, 2012.
- [231] Jonathan T. Grudin. Ai and hci: Two fields divided by a common focus. *AI Mag.*, 30:48–57, 2009.
- [232] Madeleine Grunde-McLaughlin, Michelle S. Lam, Ranjay Krishna, Daniel S. Weld, and Jeffrey Heer. Designing llm chains by adapting techniques from crowdsourcing workflows. *ArXiv*, abs/2312.11681, 2023.
- [233] Joel Grus. I don’t like notebooks. https://docs.google.com/presentation/d/1n2RlMdmv1p25Xy5thJUhkKGvjtV-dkAIsUXP-AL4ffI/edit#slide=id.g362da58057_0_1, 2018.
- [234] Xuemei Gu and Mario Krenn. Generation and human-expert evaluation of interesting research ideas using knowledge graphs and large language models. 2024.
- [235] Melody Y Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, et al. Deliberative alignment: Reasoning enables safer language models. *arXiv preprint arXiv:2412.16339*, 2024.
- [236] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [237] Jose A Guridi, Angel Hsing-Chi Hwang, Duarte Santo, Maria Goula, Cristobal Cheyre, Lee Humphreys, and Marco Rangel. From fake perfects to conversational imperfects: Exploring image-generative ai as a boundary object for participatory design of public spaces. *Proceedings of the ACM on Human-Computer Interaction*, 9(2):1–33, 2025.
- [238] Amy Gutmann and Dennis F Thompson. *Why deliberative democracy?* Princeton University Press, 2004.
- [239] Carl Gutwin and Saul Greenberg. A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work (CSCW)*, 11(3):411–446, 2002.
- [240] Margaret Hagan. Prototyping for policy. In *Legal Design*, pages 9–31. Edward Elgar Publishing, 2021.

- [241] Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. Evaluating large language models in generating synthetic hci research data: A case study. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.
- [242] Anders Hanberger. What is the policy problem? methodological challenges in policy evaluation. *Evaluation*, 7(1):45–62, 2001.
- [243] Ning Hao, Yixuan Ku, Meigui Liu, Yi Hu, Mark Bodner, Roland H. Grabner, and Andreas Fink. Reflection enhances creativity: Beneficial effects of idea evaluation on idea generation. *Brain and Cognition*, 103:30–37, 2016.
- [244] Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *ArXiv*, abs/2305.14992, 2023.
- [245] Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. *ArXiv*, abs/2305.11554, 2023.
- [246] Steve Harrison, Deborah Tatar, and Phoebe Sengers. The three paradigms of hci. In *Alt. Chi. Session at the SIGCHI Conference on human factors in computing systems San Jose, California, USA*, pages 1–18, 2007.
- [247] Robert Hart. Chatbots Can Trigger a Mental Health Crisis. What to Know About ‘AI Psychosis’. <https://time.com/7307589/ai-psychosis-chatgpt-mental-health/>, 08 2025.
- [248] Gaole He, Gianluca Demartini, and Ujwal Gadiraju. Plan-then-execute: An empirical study of user trust and team performance when using llm agents as a daily assistant. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–22, 2025.
- [249] Luxi He, Nimra Nadeem, Michel Liao, Howard Chen, Danqi Chen, Mariano-Florentino Cuéllar, and Peter Henderson. Statutory construction and interpretation for artificial intelligence. *arXiv preprint arXiv:2509.01186*, 2025.
- [250] Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S. Weld, and Marti A. Hearst. Augmenting scientific papers with just-in-time, position-sensitive definitions of terms and symbols. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2020.
- [251] Melissa Heikkilä. How AI-generated text is poisoning the internet. *MIT Technology Review*, 2022.

- [252] Kashmir Hill. A Teen Was Suicidal. ChatGPT Was the Friend He Confided In. <https://www.nytimes.com/2025/08/26/technology/chatgpt-openai-suicide.html>, 08 2025.
- [253] Kashmir Hill. They Asked an A.I. Chatbot Questions. The Answers Sent Them Spiraling. <https://www.nytimes.com/2025/06/13/technology/chatgpt-ai-chatbots-conspiracies.html>, 06 2025.
- [254] Kashmir Hill and Dylan Freedman. Chatbots Can Go Into a Delusional Spiral. Here’s How It Happens. <https://www.nytimes.com/2025/08/08/technology/ai-chatbots-delusions-chatgpt.html>, 08 2025.
- [255] Gerard P. Hodgkinson, Richard Whittington, Gerry Johnson, and Mirela Schwarz. The role of strategy workshops in strategy development processes: Formality, communication, co-ordination and inclusion. *Long Range Planning*, 39(5):479–496, 2006.
- [256] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M. Drucker. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI ’19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery.
- [257] Fred Hohman, Kanit Wongsuphasawat, Mary Beth Kery, and Kayur Patel. Understanding and visualizing data iteration in machine learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI ’20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.
- [258] James Hollan, Edwin Hutchins, and David Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(2):174–196, 2000.
- [259] Kenneth Holstein, Erik Harpstead, Rebecca Gulotta, and Jodi Forlizzi. Replay enactments: Exploring possible futures through historical data. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pages 1607–1618, 2020.
- [260] James W Hooper and Pei Hsia. Scenario-based prototyping for requirements identification. In *Proceedings of the workshop on Rapid prototyping*, pages 88–93, 1982.
- [261] Tom Hope, Doug Downey, Oren Etzioni, Daniel S. Weld, and Eric Horvitz. A computational inflection for scientific discovery. *Communications of the ACM*, 66:62 – 73, 2022.
- [262] Eric Horvitz. Principles of mixed-initiative user interfaces. In *International Conference on Human Factors in Computing Systems*, 1999.

- [263] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166, 1999.
- [264] Jeff Horwitz. Meta’s ai rules have let bots hold ‘sensual’ chats with kids, offer false medical info. <https://www.reuters.com/investigates/special-report/meta-ai-chatbot-guidelines/>, 08 2025.
- [265] Abe Bohan Hou, Hongru Du, Yichen Wang, Jingyu Zhang, Zixiao Wang, Paul Pu Liang, Daniel Khashabi, Lauren Gardner, and Tianxing He. Can a society of generative agents simulate human behavior and inform public health policy? a case study on vaccine hesitancy. *arXiv preprint arXiv:2503.09639*, 2025.
- [266] Stephanie Houde and Charles Hill. What do prototypes prototype? In *Handbook of human-computer interaction*, pages 367–381. Elsevier, 1997.
- [267] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [268] Saffron Huang, Divya Siddarth, Liane Lovitt, Thomas I Liao, Esin Durmus, Alex Tamkin, and Deep Ganguli. Collective constitutional ai: Aligning a language model with public input. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 1395–1417, 2024.
- [269] Wenlong Huang, P. Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *ArXiv*, abs/2201.07207, 2022.
- [270] Wenlong Huang, F. Xia, Ted Xiao, Harris Chan, Jacky Liang, Peter R. Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning*, 2022.
- [271] Yu Huang. Levels of ai agents: from rules to large language models. *arXiv preprint arXiv:2405.06643*, 2024.
- [272] Scott Huffman and Josh Woodward. PaLM API & MakerSuite: an approachable way to start prototyping and building generative AI applications. <https://developers.googleblog.com/2023/03/announcing-palm-api-and-makersuite.html>, 2023.
- [273] Edward Hughes, Michael Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge Shi, Tom Schaul, and Tim Rocktaschel. Open-endedness is essential for artificial superhuman intelligence. *arXiv preprint arXiv:2406.04268*, 2024.

- [274] Dave Huitema, Andrew Jordan, Stefania Munaretto, and Mikael Hildén. Policy experimentation: core concepts, political dynamics, governance and impacts. *Policy Sciences*, 51:143–159, 2018.
- [275] Faria Huq, Zora Zhiruo Wang, Frank F. Xu, Tianyue Ou, Shuyan Zhou, Jeffrey P. Bigham, and Graham Neubig. Cowpilot: A framework for autonomous and human-agent collaborative web navigation. *ArXiv*, abs/2501.16609, 2025.
- [276] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [277] Hamel Husain, Isaac Flath, and John Whitaker. Thoughts On A Month With Devin. <https://www.answer.ai/posts/2025-01-08-devin.html>, 2025.
- [278] Edwin L. Hutchins, James Hollan, and Donald A. Norman. Direct manipulation interfaces. *Hum. Comput. Interact.*, 1:311–338, 1985.
- [279] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, et al. Technology probes: inspiring design for and with families. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–24, 2003.
- [280] Chip Huyen. Agents. *huyenchip.com*, January 2025.
- [281] Chip Huyen. Agents. <https://huyenchip.com/2025/01/07/agents.html>, 2025.
- [282] IBM. Ibm watson studio - automl - ibm autoai. <https://www.ibm.com/cloud/watson-studio/autoai>, 2024.
- [283] Lujain Ibrahim, Saffron Huang, Lama Ahmad, and Markus Anderljung. Beyond static ai evaluations: advancing human interaction evaluations for llm harms and risks. *ArXiv*, abs/2405.10632, 2024.
- [284] Alexandre Yukio Ichida, Felipe Meneguzzi, and Rafael C. Cardoso. Bdi agents in natural language environments. In *Adaptive Agents and Multi-Agent Systems*, 2024.
- [285] Interaction Design Foundation. What are Prototypes? <https://www.interaction-design.org/literature/topics/prototypes>, 2025.
- [286] Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.

- [287] Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *arXiv preprint arXiv:2406.06769*, 2024.
- [288] Peter Jansen, Oyvind Tafjord, Marissa Radensky, Pao Siangliulue, Tom Hope, Bhavana Dalvi, Bodhisattwa Prasad Majumder, Daniel S. Weld, and Peter Clark. Code-scientist: End-to-end semi-automated scientific discovery with code-based experimentation. 2025.
- [289] Shagun Jhaver, Seth Frey, and Amy X Zhang. Decentralizing platform power: A design space of multi-level governance in online social platforms. *Social Media+ Society*, 9(4):20563051231207857, 2023.
- [290] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023.
- [291] Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. Promptmaker: Prompt-based prototyping with large language models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22, New York, NY, USA, 2022. Association for Computing Machinery.
- [292] Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. Promptmaker: Prompt-based prototyping with large language models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22, New York, NY, USA, 2022. Association for Computing Machinery.
- [293] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.
- [294] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *ArXiv*, abs/2310.06770, 2023.
- [295] Marina Jirotko, Charlotte P Lee, and Gary M Olson. Supporting scientific collaboration: Methods, tools and concepts. *Computer Supported Cooperative Work (CSCW)*, 22:667–715, 2013.

- [296] Matthew Jörke, Yasaman S. Sefidgar, Talie Massachi, Jina Suh, and Gonzalo Ramos. Pearl: A technology probe for machine-assisted reflection on personal data. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, IUI '23, page 902–918, New York, NY, USA, 2023. Association for Computing Machinery.
- [297] Ishika Joshi, Simra Shahid, Shreeya Manasvi Venneti, Manushree Vasu, Yantao Zheng, Yunyao Li, Balaji Krishnamurthy, and Gromit Yeuk-Yin Chan. Coprompter: User-centric evaluation of llm instruction alignment for improved prompt engineering. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pages 341–365, 2025.
- [298] Gregor Jošt, Viktor Taneski, and Sašo Karakatič. The impact of large language models on programming education and student learning outcomes. *Applied Sciences*, 14(10):4115, 2024.
- [299] Project Jupyter. Jupyter Notebook UX Survey. <https://github.com/jupyter/surveys/tree/master/surveys/2015-12-notebook-ux>, 2015.
- [300] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.
- [301] Robert A Kagan. *Adversarial legalism: The American way of law*. Harvard University Press, 2019.
- [302] Minsuk Kahng, Ian Tenney, Mahima Pushkarna, Michael Xieyang Liu, James Wexler, Emily Reif, Krystal Kallarackal, Minsuk Chang, Michael Terry, and Lucas Dixon. Llm comparator: Interactive analysis of side-by-side evaluation of large language models. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):503–513, 2024.
- [303] Emma Kallina and Jatinder Singh. Mapping the tool landscape for stakeholder involvement in participatory ai: Strengths, gaps, and future directions. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2025.
- [304] Nathan Kallus. Balanced policy evaluation and learning. *Advances in neural information processing systems*, 31, 2018.
- [305] Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Kaya Stechly, Mudit Verma, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. Llms can’t plan, but can help planning in llm-modulo frameworks. *arXiv preprint arXiv:2402.01817*, 2024.
- [306] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 3363–3372, 2011.

- [307] Hyeonsu B Kang, Joseph Chee Chang, Yongsung Kim, and Aniket Kittur. Threddy: An interactive system for personalized thread-based exploration and organization of scientific literature. *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, 2022.
- [308] Hyeonsu B Kang, Rafal Kocielnik, Andrew Head, Jiangjiang Yang, Matt Latzke, Aniket Kittur, Daniel S. Weld, Doug Downey, and Jonathan Bragg. From who you know to what you read: Augmenting scientific recommendations with implicit social networks. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022.
- [309] Hyeonsu B Kang, Xin Qian, Tom Hope, Dafna Shahaf, Joel Chan, and Aniket Kittur. Augmenting scientific creativity with an analogical search engine. *ACM Transactions on Computer-Human Interaction*, 29:1 – 36, 2022.
- [310] Hyeonsu B Kang, Sherry Wu, Joseph Chee Chang, and Aniket Kittur. Synergi: A mixed-initiative system for scholarly synthesis and sensemaking. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023.
- [311] Sayash Kapoor, Noam Kolt, and Seth Lazar. Resist platform-controlled ai agents and champion user-centric agent advocates. *arXiv preprint arXiv:2505.04345*, 2025.
- [312] Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. Ai agents that matter. *ArXiv*, abs/2407.01502, 2024.
- [313] Sayash Kapoor, Benedikt Stroebl, Zachary S Siegel, Nitya Nadgir, and Arvind Narayanan. Ai agents that matter. *arXiv preprint arXiv:2407.01502*, 2024.
- [314] Atoosa Kasirzadeh. Two types of ai existential risk: decisive and accumulative. *arXiv preprint arXiv:2401.07836*, 2024.
- [315] Majeed Kazemitabaar, Jack Williams, Ian Drosos, Tovi Grossman, Austin Z. Henley, Carina Negreanu, and Advait Sarkar. Improving steering and verification in ai-assisted data analysis with interactive task decomposition. 2024.
- [316] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers. The story in the notebook: Exploratory data science using a literate programming tool. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018.
- [317] Mary Beth Kery, Donghao Ren, Fred Hohman, Dominik Moritz, Kanit Wongsuphasawat, and Kayur Patel. mage: Fluid moves between code and graphical work in computational notebooks. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020.

- [318] Khosla Ventures. Model Designer. <https://jobs.khoslaventures.com/companies/openai/jobs/52814928-model-designer>, 2025.
- [319] Elliot Kim, Avi Garg, Kenny Peng, and Nikhil Garg. Correlated errors in large language models. *arXiv preprint arXiv:2506.07962*, 2025.
- [320] Jeongyeon Kim, Sangho Suh, Lydia B Chilton, and Haijun Xia. Metaphorian: Leveraging large language models to support extended metaphor creation for science writing. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, pages 115–135, 2023.
- [321] Joongwon Kim, Bhargavi Paranjape, Tushar Khot, and Hanna Hajishirzi. Husky: A unified, open-source language agent for multi-step reasoning. 2024.
- [322] Joy Kim, S. Sterman, Allegra Argent Beal Cohen, and Michael S. Bernstein. Mechanical novel: Crowdsourcing complex work through reflection and revision. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2016.
- [323] Stephanie Kim, Jacy Reese Anthis, and Sarah Sebo. A taxonomy of robot autonomy for human-robot interaction. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, HRI '24*, page 381–393, New York, NY, USA, 2024. Association for Computing Machinery.
- [324] Tae Soo Kim, Yoonjoo Lee, Jamin Shin, Young-Ho Kim, and Juho Kim. Evallm: Interactive evaluation of large language model prompts on user-defined criteria. In *International Conference on Human Factors in Computing Systems*, 2023.
- [325] Tae Soo Kim, Yoonjoo Lee, Jamin Shin, Young-Ho Kim, and Juho Kim. Evallm: Interactive evaluation of large language model prompts on user-defined criteria. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2024.
- [326] Lucy Kimbell and Jocelyn Bailey. Prototyping and the new spirit of policy-making. *CoDesign*, 13(3):214–226, 2017.
- [327] Gary King, Emmanuela Gakidou, Nirmala Ravishankar, Ryan T Moore, Jason Lakin, Manett Vargas, Martha María Téllez-Rojo, Juan Eugenio Hernández Ávila, Mauricio Hernández Ávila, and Héctor Hernández Llamas. A “politically robust” experimental design for public policy evaluation, with application to the mexican universal health insurance program. *Journal of Policy Analysis and Management*, 26(3):479–506, 2007.

- [328] Rodney Michael Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, Miles Crawford, Doug Downey, Jason Dunkelberger, Oren Etzioni, Rob Evans, Sergey Feldman, Joseph Gorney, David W. Graham, F.Q. Hu, Regan Huff, Daniel King, Sebastian Kohlmeier, Bailey Kuehl, Michael Langan, Daniel Lin, Haokun Liu, Kyle Lo, Jaron Lochner, Kelsey MacMillan, Tyler C. Murray, Christopher Newell, Smita R Rao, Shaurya Rohatgi, Paul Sayre, Zejiang Shen, Amanpreet Singh, Luca Soldaini, Shivashankar Subramanian, A. Tanaka, Alex D Wade, Linda M. Wagner, Lucy Lu Wang, Christopher Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Madeleine van Zuylen, and Daniel S. Weld. The semantic scholar open data platform. *ArXiv*, abs/2301.10140, 2023.
- [329] Megan Kinniment, Lucas Jun Koba Sato, Haoxing Du, Brian Goodrich, Max Hasin, Lawrence Chan, Luke Harold Miles, Tao R. Lin, Hjalmar Wijk, Joel Burget, Aaron Ho, Elizabeth Barnes, and Paul Francis Christiano. Evaluating language-model agents on realistic autonomous tasks. *ArXiv*, abs/2312.11671, 2023.
- [330] Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, et al. The prism alignment project: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. *arXiv preprint arXiv:2404.16019*, 2024.
- [331] Maaike Kleinsmann and Rianne Valkenburg. Barriers and enablers for creating shared understanding in co-design projects. *Design studies*, 29(4):369–386, 2008.
- [332] Kate Klonick. The facebook oversight board: Creating an independent institution to adjudicate online free expression. *Yale LJ*, 129:2418, 2019.
- [333] Donald Ervin Knuth. Literate programming. In *Computer/law journal*, 1984.
- [334] Verena Kontschieder. Prototyping in Policy: What For?! <https://conferences.law.stanford.edu/prototyping-for-policy/2018/10/22/prototyping-in-policy-what-for/>, 2018.
- [335] Andrew Konya, Aviv Ovadya, Kevin Feng, Quan Ze Chen, Lisa Schirch, Colin Irwin, and Amy X. Zhang. Chain of alignment: Integrating public will with expert intelligence for language model alignment, 2024.
- [336] Andrew Konya, Lisa Schirch, Colin Irwin, and Aviv Ovadya. Democratic policy development using collective dialogues and ai. *arXiv preprint arXiv:2311.02242*, 2023.
- [337] Tomek Korbak, Joshua Clymer, Benjamin Hilton, Buck Shlegeris, and Geoffrey Irving. A sketch of an ai control safety case. *arXiv preprint arXiv:2501.17315*, 2025.

- [338] Vinay Koshy, Frederick Choi, Yi-Shyuan Chiang, Hari Sundaram, Eshwar Chandrasekharan, and Karrie Karahalios. Venire: A machine learning-guided panel review system for community content moderation. *arXiv preprint arXiv:2410.23448*, 2024.
- [339] PM Krafft, Meg Young, Michael Katell, Jennifer E Lee, Shankar Narayan, Micah Epstein, Dharma Dailey, Bernease Herman, Aaron Tam, Vivian Guetler, et al. An action-oriented ai policy toolkit for technology audits by community advocates and activists. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 772–781, 2021.
- [340] Travis Kriplean, Jonathan Morgan, Deen Freelon, Alan Borning, and Lance Bennett. Supporting reflective public thought with considerit. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 265–274, 2012.
- [341] Michael Krumdick, Charles Lovering, Varshini Reddy, Seth Ebner, and Chris Tanner. No free labels: Limitations of llm-as-a-judge without human grounding. *arXiv preprint arXiv:2503.05061*, 2025.
- [342] K. P. Kruzan, Madhu C. Reddy, Jason J. Washburn, and D. Mohr. Developing a mobile app for young adults with nonsuicidal self-injury: A prototype feedback study. *International Journal of Environmental Research and Public Health*, 19, 2022.
- [343] Chinmay Kulkarni, Stefania Druga, Minsuk Chang, Alex Fiannaca, Carrie Cai, and Michael Terry. A word is worth a thousand pictures: Prompts as ai design material. *arXiv preprint arXiv:2303.12647*, 2023.
- [344] Jan Kulveit, Raymond Douglas, Nora Ammann, Deger Turan, David Krueger, and David Duvenaud. Gradual disempowerment: Systemic existential risks from incremental ai development. *arXiv preprint arXiv:2501.16946*, 2025.
- [345] Dominik Kundel. OpenAI Harmony Response Format. <https://cookbook.openai.com/articles/openai-harmony/>, August 2025.
- [346] Tzu-Sheng Kuo, Quan Ze Chen, Amy X Zhang, Jane Hsieh, Haiyi Zhu, and Kenneth Holstein. Polycraft: Supporting collaborative and participatory policy design through case-grounded deliberation. *arXiv preprint arXiv:2409.15644*, 2024.
- [347] Tzu-Sheng Kuo, Aaron Lee Halfaker, Zirui Cheng, Jiwoo Kim, Meng-Hsin Wu, Tongshuang Wu, Kenneth Holstein, and Haiyi Zhu. Wikibench: Community-driven data curation for ai evaluation on wikipedia. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–24, 2024.

- [348] Tzu-Sheng Kuo, Hong Shen, Jisoo Geum, Nev Jones, Jason I Hong, Haiyi Zhu, and Kenneth Holstein. Understanding frontline workers’ and unhoused individuals’ perspectives on ai used in homeless services. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2023.
- [349] Thinking Machines Lab. Tinker, 2025.
- [350] Isaac Lage, Andrew Ross, Samuel J Gershman, Been Kim, and Finale Doshi-Velez. Human-in-the-loop interpretability prior. *Advances in neural information processing systems*, 31, 2018.
- [351] Huiqian Lai. " please, don’t kill the only model that still feels human": Understanding the# keep4o backlash. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2026.
- [352] Michelle S Lam, Mitchell L Gordon, Danaë Metaxa, Jeffrey T Hancock, James A Landay, and Michael S Bernstein. End-user audits: A system empowering communities to lead large-scale investigations of harmful algorithmic behavior. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–34, 2022.
- [353] Michelle S Lam, Fred Hohman, Dominik Moritz, Jeffrey P Bigham, Kenneth Holstein, and Mary Beth Kery. Ai policy projector: Grounding llm policy design in iterative mapmaking. *arXiv preprint arXiv:2409.18203*, 2024.
- [354] Max Lamparth, Declan Grabb, Amy Franks, Scott Gershan, Kaitlyn N Kunstman, Aaron Lulla, Monika Drummond Roots, Manu Sharma, Aryan Shrivastava, Nina Vasan, et al. Moving beyond medical exam questions: A clinician-annotated dataset of real-world tasks and ambiguity in mental healthcare. *arXiv preprint arXiv:2502.16051*, 2025.
- [355] James Landay. “AI For Good” Isn’t Good Enough: A Call for Human-Centered AI. <https://hai.stanford.edu/events/ai-good-isnt-good-enough-call-human-centered-ai>, 2024.
- [356] Hélène Landemore and Jean-Michel Fourniau. Citizens’ assemblies, a new form of democratic representation? *Participations*, 34(3):5–36, 2022.
- [357] Hélène Landemore. Can artificial intelligence bring deliberation to the masses. *Conversations in philosophy, law, and politics*, pages 39–69, 2024.
- [358] LangChain. Human-in-the-loop. https://langchain-ai.github.io/langgraph/concepts/human_in_the_loop/, 2024.

- [359] Walter S. Lasecki, Juho Kim, Nick Rafter, Onkur Sen, Jeffrey P. Bigham, and Michael S. Bernstein. Apparition: Crowdsourced user interfaces that come to life as you sketch them. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.
- [360] Sam Lau, Ian Drosos, Julia M. Markel, and Philip J. Guo. The design space of computational notebooks: An analysis of 60 systems in academia and industry. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–11, 2020.
- [361] Carlye A Lauff, Daria Kotys-Schwartz, and Mark E Rentschler. What is a prototype? what are the roles of prototypes in companies? *Journal of Mechanical Design*, 140(6):061102, 2018.
- [362] John Law, WE Bijker, Thomas P Hughes, and Trevor Pinch. Technology and heterogeneous engineering: The case of portuguese expansion. *The social construction of technological systems: New directions in the sociology and history of technology*, 1:105–128, 2012.
- [363] Lane Lawley and Christopher J. MacLellan. Val: Interactive task learning with gpt dialog parsing. *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2024.
- [364] Seth Lazar. Legitimacy, authority, and democratic. *Oxford studies in political philosophy*, 10:28, 2024.
- [365] Charlotte P Lee. Boundary negotiating artifacts: Unbinding the routine of boundary objects and embracing chaos in collaborative work. *Computer Supported Cooperative Work (CSCW)*, 16:307–339, 2007.
- [366] Hao-Ping Lee, Advait Sarkar, Lev Tankelevitch, Ian Drosos, Sean Rintel, Richard Banks, and Nicholas Wilson. The impact of generative ai on critical thinking: Self-reported reductions in cognitive effort and confidence effects from a survey of knowledge workers. In *Proceedings of the 2025 CHI conference on human factors in computing systems*, pages 1–22, 2025.
- [367] Hyunin Lee, Jinglue Xu, Jeffrey Seely, Donghyun Lee, Matei Zaharia, and Yujin Tang. Recursive harness self-improvement. *arXiv preprint arXiv:2607.15524*, 2026.
- [368] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80, 2004.
- [369] Mina Lee, Katy Ilonka Gero, John Joon Young Chung, Simon Buckingham Shum, Vipul Raheja, Hua Shen, Subhashini Venugopalan, Thiemo Wambsganss, David Zhou,

- Emad A. Alghamdi, Tal August, Avinash Bhat, Madiha Zahrah Choksi, Senjuti Dutta, Jin L.C. Guo, Md. Naimul Hoque, Yewon Kim, Seyed Parsa Neshaei, Agnia Sergeyuk, Antonette Shibani, Disha Shrivastava, Lila Shroff, Jessi Stark, S. Sterman, Sitong Wang, Antoine Bosselut, Daniel Buschek, Joseph Chee Chang, Sherol Chen, Max Kreminski, Joonsuk Park, Roy Pea, Eugenia H. Rho, Shannon Zejiang Shen, and Pao Siangliulue. A design space for intelligent and interactive writing assistants. *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2024.
- [370] Mina Lee, Percy Liang, and Qian Yang. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–19, 2022.
- [371] Seongyun Lee, Sue Hyun Park, Seungone Kim, and Minjoon Seo. Aligning to thousands of preferences via system message generalization. *arXiv preprint arXiv:2405.17977*, 2024.
- [372] Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses. *arXiv preprint arXiv:2603.28052*, 2026.
- [373] Yoonjoo Lee, Hyeonsu B Kang, Matt Latzke, Juho Kim, Jonathan Bragg, Joseph Chee Chang, and Pao Siangliulue. Paperweaver: Enriching topical paper alerts by contextualizing recommended papers with user-collected papers. In *International Conference on Human Factors in Computing Systems*, 2024.
- [374] Pascale Lehoux, Fiona Alice Miller, and Bryn Williams-Jones. Anticipatory governance and moral imagination: Methodological insights from a scenario-based public deliberation study. *Technological Forecasting and Social Change*, 151:119800, 2020.
- [375] Germán Leiva, Nolwenn Maudet, Wendy Mackay, and Michel Beaudouin-Lafon. Enact: Reducing designer–developer breakdowns when prototyping custom interactions. 26(3), may 2019.
- [376] Jo Lenaghan. Involving the public in rationing decisions. the experience of citizens juries. *Health policy*, 49(1-2):45–61, 1999.
- [377] Paul M Leonardi. Materiality, sociomateriality, and socio-technical systems: What do these terms mean? how are they different? do we need them? *Materiality and organizing: Social interaction in a technological world*, 25(10):10–1093, 2012.
- [378] Ang Li, Yin Zhou, Vethavikashini Chithrura Raghuram, Tom Goldstein, and Micah Goldblum. Commercial llm agents are already vulnerable to simple yet dangerous attacks. *arXiv preprint arXiv:2502.08586*, 2025.

- [379] Haotian Li, Lu Ying, Haidong Zhang, Yingcai Wu, Huamin Qu, and Yun Wang. Notable: On-the-fly assistant for data storytelling in computational notebooks. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2023.
- [380] Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan Ilgen, Emma Pierson, Pang Wei W Koh, and Yulia Tsvetkov. Mediq: Question-asking llms and a benchmark for reliable interactive clinical reasoning. *Advances in Neural Information Processing Systems*, 37:28858–28888, 2024.
- [381] Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259*, 2023.
- [382] Xiang Lisa Li, Neil Chowdhury, Daniel D Johnson, Tatsunori Hashimoto, Percy Liang, Sarah Schwettmann, and Jacob Steinhardt. Eliciting language model behaviors with investigator agents. *arXiv preprint arXiv:2502.01236*, 2025.
- [383] Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*, 2024.
- [384] Jenny T. Liang, Melissa Lin, Nikitha Rao, and Brad A. Myers. Prompts are programs too! understanding how developers build software containing prompts. *Proc. ACM Softw. Eng.*, 2(FSE), June 2025.
- [385] Q Vera Liao, Hariharan Subramonyam, Jennifer Wang, and Jennifer Wortman Vaughan. Designerly understanding: Information needs for model transparency to support design ideation for ai-powered user experience. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–21, 2023.
- [386] Q Vera Liao and Jennifer Wortman Vaughan. Ai transparency in the age of llms: A human-centered research roadmap. *arXiv preprint arXiv:2306.01941*, 2023.
- [387] Zhehui Liao, Maria Antoniak, Inyoung Cheong, Evie Yu-Yen Cheng, Ai-Heng Lee, Kyle Lo, Joseph Chee Chang, and Amy X Zhang. Llms as research tools: A large scale survey of researchers’ usage and perceptions. *arXiv preprint arXiv:2411.05025*, 2024.
- [388] Henry Lieberman. Autonomous interface agents. *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, 1997.
- [389] Henry Lieberman. Autonomous interface agents. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 67–74, 1997.

- [390] Youn-Kyung Lim, Erik Stolterman, and Josh Tenenber. The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 15(2):1–27, 2008.
- [391] Phoebe Lin and Jessica Van Brummelen. Engaging teachers to co-design integrated ai curriculum for k-12 classrooms. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–12, 2021.
- [392] Yanna Lin, Haotian Li, Leni Yang, Aoyu Wu, and Huamin Qu. Inksight: Leveraging sketch interaction for documenting chart findings in computational notebooks. *IEEE Transactions on Visualization and Computer Graphics*, 30:944–954, 2023.
- [393] Liner.ai. Machine learning in a few clicks. <https://www.liner.ai/>, 2022.
- [394] Grace Liu, Brian Christian, Tsvetomira Dumbalska, Michiel A Bakker, and Rachit Dubey. Ai assistance reduces persistence and hurts independent performance. *arXiv preprint arXiv:2604.04721*, 2026.
- [395] Jiacheng Liu, Xiaohan Zhao, Xinyi Shang, and Zhiqiang Shen. Dive into claude code: The design space of today’s and future ai agent systems. *arXiv preprint arXiv:2604.14228*, 2026.
- [396] Yang Liu, Alex Kale, Tim Althoff, and Jeffrey Heer. Boba: Authoring and visualizing multiverse analyses. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1753–1763, 2020.
- [397] Yiren Liu, Si Chen, Haocong Cheng, Mengxia Yu, Xiao Ran, Andrew Mo, Yiliu Tang, and Yun Huang. How ai processing delays foster creativity: Exploring research question co-creation with an llm-based agent. In *International Conference on Human Factors in Computing Systems*, 2024.
- [398] Yue Liu, Sin Kit Lo, Qinghua Lu, Liming Zhu, Dehai Zhao, Xiwei Xu, Stefan Harrer, and Jon Whittle. Agent design pattern catalogue: A collection of architectural patterns for foundation model based agents. *arXiv preprint arXiv:2405.10467*, 2024.
- [399] Kyle Lo, Zejiang Shen, Benjamin Newman, Joseph Chee Chang, Russell Authur, Erin Bransom, Stefan Candra, Yoganand Chandrasekhar, Regan Huff, Bailey Kuehl, Amanpreet Singh, Chris Wilhelm, Angele Zamarron, Marti A. Hearst, Daniel S. Weld, Doug Downey, and Luca Soldaini. Papermage: A unified toolkit for processing, representing, and manipulating visually-rich scientific documents. In *Conference on Empirical Methods in Natural Language Processing*, 2023.
- [400] Lobe.ai. Machine Learning Made Easy. <https://www.lobe.ai/>, 2021.

- [401] Steve Lohr. A.I. Is Coming for Lawyers, Again. <https://www.nytimes.com/2023/04/10/technology/ai-is-coming-for-lawyers-again.html>, 04 2023.
- [402] Duri Long, Takeria Blunt, and Brian Magerko. Co-designing ai literacy exhibits for informal learning spaces. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–35, 2021.
- [403] Ryan Louie, Ananjan Nandi, William Fang, Cheng Chang, Emma Brunskill, and Diyi Yang. Roleplay-doh: Enabling domain-experts to create llm-simulated patients via eliciting and adhering to principles. *arXiv preprint arXiv:2407.00870*, 2024.
- [404] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.
- [405] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.
- [406] Danwei Tran Luciani, Martin Lindvall, Jonas Löwgren, et al. Machine learning as a design material: a curated collection of exemplars for visual interaction. *DS 91: Proceedings of NordDesign 2018, Linköping, Sweden, 14th-17th August 2018*, 2018.
- [407] Kai Lukoff, Ulrik Lyngs, Himanshu Zade, J Vera Liao, James Choi, Kaiyue Fan, Sean A Munson, and Alexis Hiniker. How the design of youtube influences user sense of agency. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2021.
- [408] Celia Lury and Nina Wakeford. *Inventive methods: The happening of the social*. Routledge, 2012.
- [409] Xiao Ma, Swaroop Mishra, Ariel Liu, Sophie Ying Su, Jilin Chen, Chinmay Kulkarni, Heng-Tze Cheng, Quoc V. Le, and Ed Huai hsin Chi. Beyond chatbots: Explorellm for structured thoughts and personalized model responses. *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2023.
- [410] Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, and Aixin Sun. Sciagent: Tool-augmented language models for scientific reasoning. *ArXiv*, abs/2402.11451, 2024.
- [411] Wendy E Mackay. *Users and customizable software: A co-adaptive phenomenon*. PhD thesis, Massachusetts Institute of Technology, 1990.

- [412] Wendy E Mackay. Parasitic or symbiotic? redefining our relationship with intelligent systems. In *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–2, 2024.
- [413] Wendy E Mackay and Michel Beaudouin-Lafon. Participatory design and prototyping. In *Handbook of Human Computer Interaction*, pages 1–33. Springer, 2023.
- [414] Michael A Madaio, Luke Stark, Jennifer Wortman Vaughan, and Hanna Wallach. Co-designing checklists to understand organizational challenges and opportunities around fairness in ai. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–14, 2020.
- [415] Pattie Maes. Agents that reduce work and information overload. *Commun. ACM*, 37:30–40, 1994.
- [416] Pattie Maes and Alan Wexelblat. Interface agents. *Conference Companion on Human Factors in Computing Systems*, 1996.
- [417] Narges Mahyar, Michael R James, Michelle M Ng, Reginald A Wu, and Steven P Dow. Communitycrit: inviting the public to improve and evaluate urban design ideas through micro-activities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2018.
- [418] Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Sanchaita Hazra, Ashish Sabharwal, and Peter Clark. Data-driven discovery with large generative models. *ArXiv*, abs/2402.13610, 2024.
- [419] Spyros Makridakis. The forthcoming artificial intelligence (ai) revolution: Its impact on society and firms. *Futures*, 90:46–60, 2017.
- [420] Manus. Manus AI - The AI Assistant That Turns Thoughts Into Actions. <https://www.manusai.io/>, 2025.
- [421] Noortje Marres, Michael Castelle, Beatrice Gobbo, Chiara Poletti, and James Tripp. Ai as super-controversy: Eliciting ai and society controversies with an extended expert community in the uk. *Big Data & Society*, 11(2):20539517241255103, 2024.
- [422] Michael McCurdy, Christopher Connors, Guy Pyrzak, Bob Kanefsky, and Alonso Vera. Breaking the fidelity barrier: an examination of our current characterization of prototypes and an example of a mixed-fidelity success. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1233–1242, 2006.

- [423] Claire Mellier and Rich Wilson. A Global Citizens’ Assembly on the Climate and Ecological Crisis. <https://carnegieendowment.org/research/2023/02/a-global-citizens-assembly-on-the-climate-and-ecological-crisis?lang=en>, 2023.
- [424] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.
- [425] Meta. Frontier AI Framework, February 2025.
- [426] METR. Task-Completion Time Horizons of Frontier AI Models. <https://metr.org/time-horizons/>, 2026.
- [427] Christian Meurisch, Cristina A. Mihale-Wilson, Adrian Hawlitschek, Florian Giger, Florian Müller, Oliver Hinz, and M. Mühlhäuser. Exploring user expectations of proactive ai systems. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4:1 – 22, 2020.
- [428] Ank Michels and Laurens De Graaf. Examining citizen participation: Local participatory policy making and democracy. *Local Government Studies*, 36(4):477–491, 2010.
- [429] Microsoft. AutoGen: Enabling next-generation large language model applications. <https://www.microsoft.com/en-us/research/blog/autogen-enabling-next-generation-large-language-model-applications/>, 2023.
- [430] Microsoft. System message framework and template recommendations for Large Language Models (LLMs). <https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/system-message>, 2023.
- [431] Microsoft. Agent Identity and Lifecycle. <https://microsoft.github.io/autogen/stable/user-guide/core-user-guide/core-concepts/agent-identity-and-lifecycle.html>, 2024.
- [432] Microsoft. Azure automated machine learning - automl. <https://azure.microsoft.com/en-us/products/machine-learning/automatedml/>, 2024.
- [433] Microsoft. Collaborative tools to help you create more effective and responsible human-AI experiences. <https://www.microsoft.com/en-us/haxtoolkit/>, 2024.
- [434] Microsoft. Welcome to Copilot in Word. <https://support.microsoft.com/en-us/office/welcome-to-copilot-in-word-2135e85f-a467-463b-b2f0-c51a46d625d1>, 2025.

- [435] Katharine Miller. A Moderator ChatBot for Civic Discourse. <https://hai.stanford.edu/news/moderator-chatbot-civic-discourse>, 2020.
- [436] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.
- [437] Aditi Mishra, Bretho Danzy, Utkarsh Soni, Anjana Arunkumar, Jinbin Huang, Bum Chul Kwon, and Chris Bryan. Promptaid: Visual prompt exploration, perturbation, testing and iteration for large language models. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):6946–6962, 2025.
- [438] Swati Mishra and Jeffrey M Rzeszutarski. Designing interactive transfer learning tools for ml non-experts. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.
- [439] Margaret Mitchell, Avijit Ghosh, Alexandra Sasha Luccioni, and Giada Pistilli. Fully autonomous ai agents should not be developed. *arXiv preprint arXiv:2502.02649*, 2025.
- [440] Albert C Molewijk, Tineke Abma, Margreet Stolper, and Guy Widdershoven. Teaching ethics in the clinic. the theory and practice of moral case deliberation. *Journal of Medical Ethics*, 34(2):120–124, 2008.
- [441] Robert Munro Monarch. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster, 2021.
- [442] M Jae Moon. Searching for inclusive artificial intelligence for social good: Participatory governance and policy recommendations for making ai more inclusive and benign for society. *Public Administration Review*, 83(6):1496–1505, 2023.
- [443] Jared Moore, Declan Grabb, William Agnew, Kevin Klyman, Stevie Chancellor, Desmond C Ong, and Nick Haber. Expressing stigma and inappropriate responses prevents llms from safely replacing mental health providers. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pages 599–627, 2025.
- [444] Steven Moore, Q Vera Liao, and Hariharan Subramonyam. failurenotes: Supporting designers in understanding the limits of ai models for computer vision tasks. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2023.
- [445] Kate Moran and Kelley Gordon. How to Conduct a Heuristic Evaluation. <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>, 06 2025.

- [446] Meredith Ringel Morris. Hci for agi. *Interactions*, 32(2):26–32, 2025.
- [447] Meredith Ringel Morris, Jascha Sohl-dickstein, Noah Fiedel, Tris Warkentin, Allan Dafoe, Aleksandra Faust, Clement Farabet, and Shane Legg. Levels of agi: Operationalizing progress on the path to agi. *arXiv preprint arXiv:2311.02462*, 2023.
- [448] Meredith Ringel Morris, Jascha Sohl-Dickstein, Noah Fiedel, Tris Warkentin, Allan Dafoe, Aleksandra Faust, Clement Farabet, and Shane Legg. Position: Levels of agi for operationalizing progress on the path to agi. In *Forty-first International Conference on Machine Learning*, 2024.
- [449] Iraklis Moutidis and Hywel TP Williams. Community evolution on stack overflow. *Plos one*, 16(6):e0253010, 2021.
- [450] Hussein Mozannar, Gagan Bansal, Cheng Tan, Adam Fourney, Victor Dibia, Jingya Chen, Jack Gerrits, Tyler Payne, Matheus Kunzler Maldaner, Madeleine Grundle-McLaughlin, Eric Zhu, Griffin Bassman, Jacob Alber, Peter Chang, Ricky Loynd, Friederike Niedtner, Ece Kamar, Maya Murad, Rafah Hosn, and Saleema Amershi. Magentic-ui: Towards human-in-the-loop agentic systems. *ArXiv*, abs/2507.22358, 2025.
- [451] Hussein Mozannar, Arvind Satyanarayan, and David Sontag. Teaching humans when to defer to a classifier via exemplars. In *Proceedings of the aaai conference on artificial intelligence*, volume 36, pages 5323–5331, 2022.
- [452] Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International conference on machine learning*, pages 7076–7087. PMLR, 2020.
- [453] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.
- [454] Cleo Nardo. The Waluigi Effect (mega-post). *Less Wrong*, 2023.
- [455] Neuronic AI. AutoGPT-UI: An interface for managing AutoGPT agents, 2024.
- [456] Benjamin Newman, Yoonjoo Lee, Aakanksha Naik, Pao Siangliulue, Raymond Fok, Juho Kim, Daniel S Weld, Joseph Chee Chang, and Kyle Lo. Arxivdigestables: Synthesizing scientific literature into tables using language models. *arXiv preprint arXiv:2410.22360*, 2024.

- [457] Felicia Ng, Jina Suh, and Gonzalo Ramos. Understanding and supporting knowledge decomposition for machine teaching. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, DIS '20, page 1183–1194, New York, NY, USA, 2020. Association for Computing Machinery.
- [458] Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective. *arXiv preprint arXiv:2209.00626*, 2022.
- [459] Tien N Nguyen and Raymond Choo. Human-in-the-loop xai-enabled vulnerability detection, investigation, and mitigation. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1210–1212. IEEE, 2021.
- [460] Jakob Nielsen. Finding usability problems through heuristic evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 373–380, 1992.
- [461] Jakob Nielsen. Intent by Discovery: Designing the AI User Experience. <https://www.uxtigers.com/post/intent-ux>, 2026.
- [462] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, 1990.
- [463] Harshit Nigam, Manasi S. Patwardhan, Lovekesh Vig, and Gautam M. Shroff. Acceleron: A tool to accelerate research ideation. *ArXiv*, abs/2403.04382, 2024.
- [464] Jędrzej Niklas and Lina Dencik. What rights matter? examining the place of social rights in the eu’s artificial intelligence policy debate. *Internet Policy Review*, 10(3), 2021.
- [465] Stefanos Nikolaidis, Yu Xiang Zhu, David Hsu, and Siddhartha S. Srinivasa. Human-robot mutual adaptation in shared autonomy. *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 294–302, 2017.
- [466] André Nogueira and Ruth Schmidt. Participatory policy design: igniting systems change through prototyping. *Policy Design and Practice*, 5(1):32–50, 2022.
- [467] Don Norman. *Emotional design: Why we love (or hate) everyday things*. Basic books, 2007.
- [468] Donald A Norman. Cognitive engineering. In *User centered system design*, pages 31–62. CRC Press, 1986.

- [469] Donald A Norman and Stephen W Draper. *User centered system design; new perspectives on human-computer interaction*. L. Erlbaum Associates Inc., 1986.
- [470] Changhoon Oh, Seonghyeon Kim, Jinhan Choi, Jinsu Eun, Soomin Kim, Juho Kim, Joonhwan Lee, and Bongwon Suh. Understanding how people reason about aesthetic evaluations of artificial intelligence. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, DIS '20, page 1169–1181, New York, NY, USA, 2020. Association for Computing Machinery.
- [471] Jasper Tran O’Leary, Gabrielle Benabdallah, and Nadya Peek. Imprimer: Computational notebooks for cnc milling. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.
- [472] Gary M Olson and Judith S Olson. Distance matters. *Human-computer interaction*, 15(2-3):139–178, 2000.
- [473] Santiago Ontanón and Enric Plaza. Arguments and counterexamples in case-based joint deliberation. In *International Workshop on Argumentation in Multi-Agent Systems*, pages 36–53. Springer, 2006.
- [474] OpenAI. Custom instructions for ChatGPT. <https://openai.com/blog/custom-instructions-for-chatgpt>, 2023.
- [475] OpenAI. Democratic inputs to AI. <https://openai.com/index/democratic-inputs-to-ai/>, 2023.
- [476] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.
- [477] OpenAI. GPT Models. <https://platform.openai.com/docs/guides/gpt>, 2023.
- [478] OpenAI. Introducing GPTs. <https://openai.com/blog/introducing-gpts>, 2023.
- [479] OpenAI. ChatGPT. <https://openai.com/chatgpt>, 2024.
- [480] OpenAI. Introducing canvas. <https://openai.com/index/introducing-canvas/>, 2024.
- [481] OpenAI. Introducing the Model spec. <https://openai.com/index/introducing-the-model-spec/>, 2024.
- [482] OpenAI. Learning to Reason with LLMs. <https://openai.com/index/learning-to-reason-with-llms/>, 2024.

- [483] OpenAI. Memory and new controls for ChatGPT. <https://openai.com/index/memory-and-new-controls-for-chatgpt/>, 2024.
- [484] OpenAI. Playground. <https://platform.openai.com/playground>, 2024.
- [485] OpenAI. Collective alignment: public input on our Model Spec. <https://openai.com/index/collective-alignment-aug-2025-updates/>, 08 2025.
- [486] OpenAI. Computer-Using Agent, 2025.
- [487] OpenAI. Customizing Your ChatGPT Personality. <https://help.openai.com/en/articles/11899719-customizing-your-chatgpt-personality>, 2025.
- [488] OpenAI. Introducing deep research. <https://openai.com/index/introducing-deep-research/>, 2025.
- [489] OpenAI. Introducing GPT-4.1 in the API. <https://openai.com/index/gpt-4-1/>, 04 2025.
- [490] OpenAI. Introducing Operator. <https://openai.com/index/introducing-operator/>, 2025.
- [491] OpenAI. OpenAI Model Spec. <https://model-spec.openai.com/2025-04-11.html>, 04 2025.
- [492] OpenAI. Operator System Card. https://cdn.openai.com/operator_system_card.pdf, 2025.
- [493] OpenAI. What we're optimizing ChatGPT for. <https://openai.com/index/how-we-re-optimizing-chatgpt/>, 08 2025.
- [494] OpenAI Platform. Assistants API. <https://platform.openai.com/docs/assistants/overview>, 2024.
- [495] Wanda J Orlikowski. Sociomaterial practices: Exploring technology at work. *Organization studies*, 28(9):1435–1448, 2007.
- [496] Wanda J. Orlikowski. The sociomateriality of organisational life: considering technology in management research. *Cambridge Journal of Economics*, 34:125–141, 2010.
- [497] Elinor Ostrom. *Governing the commons: The evolution of institutions for collective action*. Cambridge university press, 1990.

- [498] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [499] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022.
- [500] Aviv Ovadya, Kyle Redman, Luke Thorburn, Quan Ze Chen, Oliver Smith, Flynn Devine, Andrew Konya, Smitha Milli, Manon Revel, Kevin Feng, et al. Position: Democratic ai is possible. the democracy levels framework shows how it might work. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025.
- [501] James C Overholser. Elements of the socratic method: I. systematic questioning. *Psychotherapy: Theory, Research, Practice, Training*, 30(1):67, 1993.
- [502] Shabnam Ozlati and Roman Yampolskiy. The formalization of ai risk management and safety standards. In *AAAI Workshops*, 2017.
- [503] p5.js. p5.js Showcase. <https://showcase.p5js.org/>, 2022.
- [504] Christine A Padesky. Socratic questioning: Changing minds or guiding discovery. In *A keynote address delivered at the European Congress of Behavioural and Cognitive Therapies, London*, volume 24, 1993.
- [505] Srishti Palani, Aakanksha Naik, Doug Downey, Amy X. Zhang, Jonathan Bragg, and Joseph Chee Chang. Relatedly: Scaffolding literature reviews with existing related work sections. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.
- [506] Christina A Pan, Sahil Yakhmi, Tara P Iyer, Evan Strasnick, Amy X Zhang, and Michael S Bernstein. Comparing the perceived legitimacy of content moderation processes: Contractors, algorithms, expert panels, and digital juries. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW1):1–31, 2022.
- [507] Rock Yuren Pang, Sebastin Santy, René Just, and Katharina Reinecke. Blip: facilitating the exploration of undesirable consequences of digital technologies. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2024.

- [508] Aditya G. Parameswaran, Shreya Shankar, Parth Asawa, Naman Jain, and Yujie Wang. Revisiting prompt engineering via declarative crowdsourcing. *ArXiv*, abs/2308.03854, 2023.
- [509] Raja Parasuraman, Thomas B Sheridan, and Christopher D Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, 30(3):286–297, 2000.
- [510] Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pages 1–18, 2022.
- [511] Chris Parnin, Gustavo Soares, Rahul Pandita, Sumit Gulwani, Jessica Rich, and Austin Z Henley. Building your own product copilot: Challenges, opportunities, and needs. *arXiv preprint arXiv:2312.14231*, 2023.
- [512] Participedia. Ireland Participatory Democracy Pilot ‘We the Citizens’. <https://participedia.net/case/1251>, 2025.
- [513] Paz Perez. The rise of the Model Designer. <https://uxdesign.cc/the-rise-of-the-model-designer-cef429d9c134>, 2024.
- [514] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.
- [515] Yi-Hao Peng, Dingzeyu Li, Jeffrey P. Bigham, and Amy Pavel. Morae: Proactively pausing ui agents for user choices. 2025.
- [516] People Powered. Participatory Policymaking. <https://www.peoplepowered.org/participatory-policymaking>, 2025.
- [517] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.
- [518] Sarah Perez. Sam Altman warns there’s no legal confidentiality when using ChatGPT as a therapist. <https://techcrunch.com/2025/07/25/sam-altman-warns-theres-no-legal-confidentiality-when-using-chatgpt-as-a-therapist/>, July 2025.
- [519] Perplexity. Perplexity AI. <https://www.perplexity.ai/>, 2024.

- [520] Perplexity. Introducing Perplexity Deep Research. <https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research>, 2025.
- [521] Savvas Petridis, Michael Terry, and Carrie J Cai. Promptinfuser: How tightly coupling ai and ui design impacts designers’ workflows. *arXiv preprint arXiv:2310.15435*, 2023.
- [522] Savvas Petridis, Michael Terry, and Carrie Jun Cai. Promptinfuser: Bringing user interface mock-ups to life with large language models. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA ’23, New York, NY, USA, 2023. Association for Computing Machinery.
- [523] Savvas Petridis, Ben Wedin, Ann Yuan, James Wexler, and Nithum Thain. Constitutionalexperts: Training a mixture of principle-based prompts. *arXiv preprint arXiv:2403.04894*, 2024.
- [524] Savvas Petridis, Benjamin D Wedin, James Wexler, Mahima Pushkarna, Aaron Donsbach, Nitesh Goyal, Carrie J Cai, and Michael Terry. Constitutionmaker: Interactively critiquing large language models by converting feedback into principles. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, pages 853–868, 2024.
- [525] Sundar Pichai. An important next step on our ai journey. <https://blog.google/technology/ai/bard-google-ai-search-updates/>, 2023.
- [526] Trevor J Pinch and Wiebe E Bijker. The social construction of facts and artefacts: Or how the sociology of science and the sociology of technology might benefit each other. *Social studies of science*, 14(3):399–441, 1984.
- [527] Ori Press, Andreas Hochlehnert, Ameya Prabhu, Vishaal Udandaraao, Ofir Press, and Matthias Bethge. Citeme: Can language models accurately cite scientific claims?, 2024.
- [528] Monroe E Price and Joshua M Price. Building legitimacy in the absence of the state: Reflections on the facebook oversight board. *International Journal of Communication*, 17:11, 2023.
- [529] Jason Proctor. Air Canada found liable for chatbot’s bad advice on plane tickets. <https://www.cbc.ca/news/canada/british-columbia/air-canada-chatbot-lawsuit-1.7116416>, 2024.
- [530] Project Let’s Talk Privacy. Policy Prototyping Guide. <https://letstalkprivacy.media.mit.edu/ltp-prototyping-guide.pdf>, 2020.

- [531] Kevin Pu, K. J. Kevin Feng, Tovi Grossman, Tom Hope, Bhavana Dalvi, Matt Latzke, Jonathan Bragg, Joseph Chee Chang, and Pao Siangliulue. Ideasynt: Iterative research idea development through evolving and composing idea facets with literature-grounded feedback. *ArXiv*, abs/2410.04025, 2024.
- [532] Kevin Pu, KJ Feng, Tovi Grossman, Tom Hope, Bhavana Dalvi Mishra, Matt Latzke, Jonathan Bragg, Joseph Chee Chang, and Pao Siangliulue. Ideasynt: Iterative research idea development through evolving and composing idea facets with literature-grounded feedback. *arXiv preprint arXiv:2410.04025*, 2024.
- [533] Kevin Pu, KJ Kevin Feng, Tovi Grossman, Tom Hope, Bhavana Dalvi Mishra, Matt Latzke, Jonathan Bragg, Joseph Chee Chang, and Pao Siangliulue. Ideasynt: Iterative research idea development through evolving and composing idea facets with literature-grounded feedback. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–31, 2025.
- [534] Public Participation Guide: Citizen Juries. United States Environmental Protection Agency. <https://www.epa.gov/international-cooperation/public-participation-guide-citizen-juries>, 2025.
- [535] Kun Qian, Lucian Popa, and Prithviraj Sen. Systemer: A human-in-the-loop system for explainable entity resolution. *Proceedings of the VLDB Endowment*, 12(12):1794–1797, 2019.
- [536] Yujia Qin, Shi Liang, Yining Ye, Kunlun Zhu, Lan Yan, Ya-Ting Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Marc H. Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. *ArXiv*, abs/2307.16789, 2023.
- [537] Quan Ze (Jim) Chen and Kevin Feng and Inyoung Cheong and Amy X. Zhang and King Xia. Case Law for AI Policy. <https://social.cs.washington.edu/case-law-ai-policy/>, 2023.
- [538] Angelica Quicksey and Chris Meierling. Policy Prototypes: How designers and policy practitioners can use prototypes to get feedback and iterate on policy. <https://designmuseumfoundation.org/policy-prototypes/>, 2022.
- [539] Napol Rachatasumrit, Jonathan Bragg, Amy X. Zhang, and Daniel S. Weld. Citeread: Integrating localized citation contexts into scientific paper reading. *27th International Conference on Intelligent User Interfaces*, 2022.
- [540] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

- [541] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *The Journal of Machine Learning Research*, 7:1655–1686, 2006.
- [542] Gonzalo Ramos, Christopher Meek, Patrice Simard, Jina Suh, and Soroush Ghorashi. Interactive machine teaching: a human-centered approach to building machine-learned models. *Human-Computer Interaction*, 35(5-6):413–451, 2020.
- [543] Anand S Rao and Michael P Georgeff. Modeling rational agents within a bdi-architecture. *Readings in agents*, pages 317–328, 1997.
- [544] Remesh Inc. Remesh: Insights with Depth, Speed, and Quality. <https://www.remesh.ai/>, 2025.
- [545] Donghao Ren, Saleema Amershi, Bongshin Lee, Jina Suh, and Jason D Williams. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE transactions on visualization and computer graphics*, 23(1):61–70, 2016.
- [546] Replit. Turn your ideas into apps with ai, 2025.
- [547] Reworkd. AgentGPT. <https://agentgpt.reworkd.ai/>, 2024.
- [548] Brandon Reynante, Steven P Dow, and Narges Mahyar. A framework for open civic design: Integrating public participation, crowdsourcing, and design thinking. *Digital Government: Research and Practice*, 2(4):1–22, 2021.
- [549] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics.
- [550] Mark O Riedl. Human-centered artificial intelligence and machine learning. *Human behavior and emerging technologies*, 1(1):33–36, 2019.
- [551] Samantha Robertson, Zijie J Wang, Dominik Moritz, Mary Beth Kery, and Fred Hohman. Angler: Helping machine translation practitioners prioritize model improvements. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2023.
- [552] Mike Robinson. Double-level languages and co-operative working. *AI & society*, 5(1):34–60, 1991.
- [553] Kyle Robison. Agents are the future AI companies promise — and desperately need. <https://www.theverge.com/2024/10/10/24266333/ai-agents-assistants-openai-google-deepmind-bots>, 2024.

- [554] Kylie Robison. Meta gets caught gaming AI benchmarks with Llama 4. <https://www.theverge.com/meta/645012/meta-llama-4-maverick-benchmarks-gaming>, 04 2025.
- [555] Erica Robles and Mikael Wiberg. Texturing the "material turn" in interaction design. In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '10, page 137–144, New York, NY, USA, 2010. Association for Computing Machinery.
- [556] Kevin Roose. Bing's A.I. Chat: 'I Want to Be Alive.'. <https://www.nytimes.com/2023/02/16/technology/bing-chatbot-transcript.html>, 2023.
- [557] Stephanie Rosenbaum, Gilbert Cockton, Kara Coyne, Michael Muller, and Thyra Rauch. Focus groups in hci: wealth of information or waste of resources? In *CHI'02 extended abstracts on human factors in computing systems*, pages 702–703, 2002.
- [558] Teddy Rosenbluth. This Therapist Helped Clients Feel Better. It Was A.I. <https://www.nytimes.com/2025/04/15/health/ai-therapist-mental-health.html>, 04 2025.
- [559] Daniela K Rosner, Saba Kawas, Wenqi Li, Nicole Tilly, and Yi-Chen Sung. Out of time, out of place: Reflections on design workshops as a research method. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1131–1141, 2016.
- [560] Jim Rudd, Ken Stern, and Scott Isensee. Low vs. high-fidelity prototyping debate. *interactions*, 3(1):76–85, 1996.
- [561] Adam Rule, Aurélien Tabard, and James Hollan. Exploration and explanation in computational notebooks. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018.
- [562] R.K. Rushmer, D.J. Hunter, and A. Steven. Using interactive workshops to prompt knowledge exchange: a realist evaluation of a knowledge to action initiative. *Public Health*, 128(6):552–560, 2014.
- [563] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.
- [564] David M Ryfe. Does deliberative democracy work? *Annu. Rev. Polit. Sci.*, 8(1):49–71, 2005.
- [565] SAE International. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. https://www.sae.org/standards/content/j3016_202104/, 2021.

- [566] Niloufar Salehi, Jaime Teevan, Shamsi T. Iqbal, and Ece Kamar. Communicating context to the crowd for complex writing tasks. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2017.
- [567] Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. In-context impersonation reveals large language models’ strengths and biases. *Advances in Neural Information Processing Systems*, 36, 2024.
- [568] Elizabeth B-N Sanders and Pieter Jan Stappers. Co-creation and the new landscapes of design. *Co-design*, 4(1):5–18, 2008.
- [569] Mark S Sanders and Ernest James McCormick. Human factors in engineering and design, 1998.
- [570] Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cino Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? *arXiv preprint arXiv:2303.17548*, 2023.
- [571] Advait Sarkar. Ai should challenge, not obey. *Communications of the ACM*, 2024.
- [572] Arvind Satyanarayan. Intelligence as agency. In *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–3, 2024.
- [573] Mike Schackermann, Graeme Beaton, Minahz Habib, Andrew Lim, Kate Larson, and Edith Law. Understanding expert disagreement in medical data analysis through structured adjudication. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–23, 2019.
- [574] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.
- [575] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *ArXiv*, abs/2302.04761, 2023.
- [576] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*, 2025.
- [577] Kjeld Schmidt and Liam Bannon. Taking cscw seriously: Supporting articulation work. *Computer supported cooperative work (CSCW)*, 1(1):7–40, 1992.

- [578] Kjeld Schmidt and Ina Wagner. Coordinative artifacts in architectural practice. In *COOP*, volume 2, pages 257–274. Citeseer, 2002.
- [579] Scite. AI for research. <https://scite.ai/>, 2025.
- [580] Andrew D. Selbst, Danah Boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. Fairness and abstraction in sociotechnical systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* '19*, page 59–68, New York, NY, USA, 2019. Association for Computing Machinery.
- [581] Mario Selvaggio, Marco Cognetti, Stefanos Nikolaidis, Serena Ivaldi, and Bruno Siciliano. Autonomy in physical human-robot interaction: A brief survey. *IEEE Robotics and Automation Letters*, 6(4):7989–7996, 2021.
- [582] Burr Settles. From theories to queries: Active learning in practice. In *Active learning and experimental design workshop in conjunction with AISTATS 2010*, pages 1–18. JMLR Workshop and Conference Proceedings, 2011.
- [583] Burr Settles. Active learning. *Synthesis lectures on artificial intelligence and machine learning*, 6(1):1–114, 2012.
- [584] Saqib Shah. Snapchat’s My AI chatbot is making people paranoid as it ‘knows your current location’. <https://www.standard.co.uk/tech/snapchat-my-ai-chatbot-making-people-paranoid-b1076287.html>, 2023.
- [585] Omar Shaikh, Shardul Sapkota, Shan Rizvi, Eric Horvitz, Joon Sung Park, Diyi Yang, and Michael S Bernstein. Creating general user models from computer use. *arXiv preprint arXiv:2505.10831*, 2025.
- [586] Shreya Shankar, JD Zamfirescu-Pereira, Björn Hartmann, Aditya Parameswaran, and Ian Arawjo. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14, 2024.
- [587] Yijia Shao, Tianshi Li, Weiyan Shi, Yanchen Liu, and Diyi Yang. PrivacyLens: Evaluating privacy norm awareness of language models in action. *arXiv preprint arXiv:2409.00138*, 2024.
- [588] Yijia Shao, Vinay Samuel, Yucheng Jiang, John Yang, and Diyi Yang. Collaborative gym: A framework for enabling and evaluating human-agent collaboration. *arXiv preprint arXiv:2412.15701*, 2024.
- [589] Yijia Shao, Vinay Samuel, Yucheng Jiang, John Yang, and Diyi Yang. Collaborative gym: A framework for enabling and evaluating human-agent collaboration. 2024.

- [590] Yonadav Shavit, Sandhini Agarwal, Miles Brundage, Steven Adler, Cullen O’Keefe, Rosie Campbell, Teddy Lee, Pamela Mishkin, Tyna Eloundou, Alan Hickey, et al. Practices for governing agentic ai systems. *Research Paper, OpenAI, December, 2023*.
- [591] Renee Shelby, Shalaleh Rismeni, Kathryn Henne, AJung Moon, Negar Rostamzadeh, Paul Nicholas, N’Mah Yilla-Akbari, Jess Gallegos, Andrew Smart, Emilio Garcia, et al. Sociotechnical harms of algorithmic systems: Scoping a taxonomy for harm reduction. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 723–741, 2023.
- [592] Hong Shen, Alicia DeVos, Motahhare Eslami, and Kenneth Holstein. Everyday algorithm auditing: Understanding the power of everyday users in surfacing harmful algorithmic behaviors. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–29, 2021.
- [593] Hua Shen, Tiffany Knearem, Reshmi Ghosh, Kenan Alkiek, Kundan Krishna, Yachuan Liu, Ziqiao Ma, Savvas Petridis, Yi-Hao Peng, Li Qiwei, et al. Towards bidirectional human-ai alignment: A systematic review for clarifications, framework, and future directions. *arXiv preprint arXiv:2406.09264*, 2024.
- [594] Judy Hanwen Shen and Alex Tamkin. How ai impacts skill formation. *arXiv preprint arXiv:2601.20245*, 2026.
- [595] Shannon Zejiang Shen, Valerie Chen, Ken Gu, Alexis Ross, Zixian Ma, Jillian Ross, Alex Gu, Chenglei Si, Wayne Chi, Andi Peng, et al. Completion $\neq$ collaboration: Scaling collaborative effort with agents. In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*.
- [596] Quan Shi, Michael Tang, Karthik Narasimhan, and Shunyu Yao. Can language models solve olympiad programming? *ArXiv*, abs/2404.10952, 2024.
- [597] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. NIPS ’23, Red Hook, NY, USA, 2023. Curran Associates Inc.
- [598] Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Neural Information Processing Systems*, 2023.
- [599] Shneiderman. Direct manipulation: A step beyond programming languages. *Computer*, 16(8):57–69, 1983.
- [600] Ben Shneiderman. Human-centered artificial intelligence: Three fresh ideas. *AIS Transactions on Human-Computer Interaction*, 12(3):109–124, 2020.

- [601] Ben Shneiderman. *Human-centered AI*. Oxford University Press, 2022.
- [602] Ben Shneiderman and Pattie Maes. Direct manipulation vs. interface agents. *Interactions*, 4:42–61, 1997.
- [603] Wesley Shrum, Joel Genuth, and Ivan Chompalov. *Structures of scientific collaboration*. MIT Press, 2007.
- [604] Tanjal Shukla, Kevin Feng, Leijie Wang, Mohammad Rostami, and Amy Zhang. Hedwig: Dynamic autonomy for coding agents under local oversight. In *Proceedings of the ACM Conference on AI and Agentic Systems*, pages 1293–1299, 2026.
- [605] Mona Sloane, Emanuel Moss, Olaitan Awomolo, and Laura Forlano. Participation is not a design fix for machine learning. In *Proceedings of the 2nd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–6, 2022.
- [606] Christopher Small, Michael Bjorkegren, Timo Erkkilä, Lynette Shaw, and Colin Megill. Polis: Scaling deliberation by mapping high dimensional opinion spaces. *Recerca: revista de pensament i anàlisi*, 26(2), 2021.
- [607] Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In *23rd International Conference on Intelligent User Interfaces*, IUI ’18, page 293–304, New York, NY, USA, 2018. Association for Computing Machinery.
- [608] Graham Smith and Corinne Wales. Citizens’ juries and deliberative democracy. *Political studies*, 48(1):51–65, 2000.
- [609] Irene Solaiman, Zeerak Talat, William Agnew, Lama Ahmad, Dylan Baker, Su Lin Blodgett, Canyu Chen, Hal Daumé III, Jesse Dodge, Isabella Duan, et al. Evaluating the social impact of generative ai systems in systems and society. *arXiv preprint arXiv:2306.05949*, 2023.
- [610] Nouran Soliman, Hyeonsu B Kang, Matt Latzke, Jonathan Bragg, Joseph Chee Chang, Amy X. Zhang, and David R. Karger. Mitigating barriers to public social interaction with meronymous communication. In *International Conference on Human Factors in Computing Systems*, 2024.
- [611] Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L Gordon, Niloofar Miresghalah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, et al. Position: A roadmap to pluralistic alignment. In *Forty-first International Conference on Machine Learning*.

- [612] Taylor Sorensen, Jared Moore, Jillian R. Fisher, Mitchell Gordon, Niloofar Miresghalah, Christopher Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. A roadmap to pluralistic alignment. *ArXiv*, abs/2402.05070, 2024.
- [613] Tobin South, Samuele Marro, Thomas Hardjono, Robert Mahari, Cedric Deslandes Whitney, Dazza Greenwood, Alan Chan, and Alex Pentland. Authenticated delegation and authorized ai agents. *arXiv preprint arXiv:2501.09674*, 2025.
- [614] Tobin South, Samuele G. Marro, Thomas Hardjono, Robert Mahari, Cedric Deslandes Whitney, Dazza Greenwood, Alan Chan, and Alex 'Sandy' Pentland. Authenticated delegation and authorized ai agents. *ArXiv*, abs/2501.09674, 2025.
- [615] Colette Stallbaumer. Introducing Microsoft 365 Copilot—A whole new way to work. <https://www.microsoft.com/en-us/microsoft-365/blog/2023/03/16/introducing-microsoft-365-copilot-a-whole-new-way-to-work/>, 2023.
- [616] Stanford Encyclopedia of Philosophy. Agency. <https://plato.stanford.edu/entries/agency/>, 2015.
- [617] Susan Leigh Star and James R Griesemer. Institutional ecology, translations' and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-39. *Social studies of science*, 19(3):387–420, 1989.
- [618] Leon Staufer, Kevin Feng, Kevin Wei, Luke Bailey, Yawen Duan, Mick Yang, A Pinar Ozisik, Stephen Casper, and Noam Kolt. The 2025 ai agent index: Documenting technical and safety features of deployed agentic ai systems. In *The 2026 ACM Conference on Fairness, Accountability, and Transparency*, pages 1536–1576, 2026.
- [619] Amanda Stent and James F. Allen. Trains-96 system evaluation. 1997.
- [620] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.
- [621] Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. *arXiv preprint arXiv:2410.12877*, 2024.
- [622] Hendrik Strobelt, Albert Webson, Victor Sanh, Benjamin Hoover, Johanna Beyer, Hanspeter Pfister, and Alexander M. Rush. Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1146–1156, 2023.

- [623] Hariharan Subramonyam, Jane Im, Colleen Seifert, and Eytan Adar. Solving separation-of-concerns problems in collaborative design of human-ai systems through leaky abstractions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.
- [624] Hariharan Subramonyam, Jane Im, Colleen Seifert, and Eytan Adar. Solving separation-of-concerns problems in collaborative design of human-ai systems through leaky abstractions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2022.
- [625] Hariharan Subramonyam, Colleen Seifert, and Eytan Adar. Protoai: Model-informed prototyping for ai-powered interfaces. In *26th International Conference on Intelligent User Interfaces*, IUI '21, page 48–58, New York, NY, USA, 2021. Association for Computing Machinery.
- [626] Hariharan Subramonyam, Colleen Seifert, and Eytan Adar. Towards a process model for co-creating ai experiences. In *Designing Interactive Systems Conference 2021*, DIS '21, page 1529–1543, New York, NY, USA, 2021. Association for Computing Machinery.
- [627] Lucille Alice Suchman. *Plans and situated actions: The problem of human-machine communication*. Cambridge university press, 1987.
- [628] Sangho Suh, Hai Dang, Ryan Yen, Josh M. Pollock, Ian Arawjo, Rubaiat Habib Kazi, Hariharan Subramonyam, Jingyi Li, Nazmus Saquib, and Arvindmani Satyanarayan. Dynamic abstractions: Building the next generation of cognitive tools and interfaces. In *ACM Symposium on User Interface Software and Technology*, 2024.
- [629] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. Sensecape: Enabling multi-level exploration and sensemaking with large language models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery.
- [630] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. Sensecape: Enabling multi-level exploration and sensemaking with large language models. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023.
- [631] Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. *ArXiv*, abs/2309.02427, 2023.
- [632] Jiao Sun, Q Vera Liao, Michael Muller, Mayank Agarwal, Stephanie Houde, Kartik Talamadupula, and Justin D Weisz. Investigating explainability of generative ai for code through scenario-based design. In *27th International Conference on Intelligent User Interfaces*, pages 212–228, 2022.

- [633] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36, 2024.
- [634] Harini Suresh, Emily Tseng, Meg Young, Mary Gray, Emma Pierson, and Karen Levy. Participation in the age of foundation models. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 1609–1621, 2024.
- [635] Don R. Swanson. Undiscovered public knowledge. *The Library Quarterly*, 56:103 – 118, 1986.
- [636] Annalisa Szymanski, Noah Ziemis, Heather A Eicher-Miller, Toby Jia-Jun Li, Meng Jiang, and Ronald A Metoyer. Limitations of the llm-as-a-judge approach for evaluating llm outputs in expert knowledge tasks. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pages 952–966, 2025.
- [637] Xiangru Tang, Qiao Jin, Kunlun Zhu, Tongxin Yuan, Yichi Zhang, Wangchunshu Zhou, Meng Qu, Yilun Zhao, Jian Tang, Zhuosheng Zhang, et al. Prioritizing safeguarding over autonomy: Risks of llm agents for science. *arXiv preprint arXiv:2402.04247*, 2024.
- [638] AIME Planning Team. Artificial intelligence measurement and evaluation at the national institute of standards and technology. *National Institute of Standards and Technology*, 2021.
- [639] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.
- [640] Michael Terry, Chinmay Kulkarni, Martin Wattenberg, Lucas Dixon, and Meredith Ringel Morris. Ai alignment in the design of interactive ai: Specification alignment, process alignment, and evaluation support. *arXiv preprint arXiv:2311.00710*, 2023.
- [641] Michael Terry, Chinmay Kulkarni, Martin Wattenberg, Lucas Dixon, and Meredith Ringel Morris. Interactive ai alignment: Specification, process, and evaluation alignment. 2023.
- [642] Michael Terry, Chinmay Kulkarni, Martin Wattenberg, Lucas Dixon, and Meredith Ringel Morris. Interactive ai alignment: Specification, process, and evaluation alignment, 2024.

- [643] Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. In Ofir Arviv, Miruna Clinciu, Kaustubh Dhole, Rotem Dror, Sebastian Gehrmann, Eliya Habba, Itay Itzhak, Simon Mille, Yotam Perlitz, Enrico Santus, João Sedoc, Michal Shmueli Scheuer, Gabriel Stanovsky, and Oyvind Tafjord, editors, *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 404–430, Vienna, Austria and virtual meeting, July 2025. Association for Computational Linguistics.
- [644] Anna Tong and Jeffrey Dastin. Insight: Race towards 'autonomous' AI agents grips Silicon Valley. <https://www.reuters.com/technology/race-towards-autonomous-ai-agents-grips-silicon-valley-2023-07-17/>, 2024.
- [645] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [646] Khai N Truong, Gillian R Hayes, and Gregory D Abowd. Storyboarding: an empirical determination of best practices and effective guidelines. In *Proceedings of the 6th conference on Designing Interactive systems*, pages 12–21, 2006.
- [647] Emily Tseng, Meg Young, Marianne Aubin Le Quéré, Aimee Rinehart, and Harini Suresh. "ownership, not just happy talk": Co-designing a participatory large language model for journalism. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pages 3119–3130, 2025.
- [648] Tiffany Tseng, Jennifer King Chen, Mona Abdelrahman, Mary Beth Kery, Fred Hohman, Adriana Hilliard, and R Benjamin Shapiro. Collaborative machine learning model building with families using co-ml. In *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*, pages 40–51, 2023.
- [649] Alan Mathison Turing. Mind. *Mind*, 59(236):433–460, 1950.
- [650] Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.
- [651] usability.gov. System usability scale (sus). <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>, 2022.
- [652] Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models - a critical investigation. *ArXiv*, abs/2305.15771, 2023.

- [653] Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005, 2023.
- [654] Colin van Noordt and Gianluca Misuraca. Artificial intelligence for the public sector: results of landscaping the use of ai in government across the european union. *Government Information Quarterly*, 39(3):101714, 2022.
- [655] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [656] Vercel. Introducing AI SDK 3.0 with Generative UI support. <https://vercel.com/blog/ai-sdk-3-generative-ui>, 2024.
- [657] Fernanda B Viégas, Martin Wattenberg, and Matthew M McKeon. The hidden order of wikipedia. In *Online Communities and Social Computing: Second International Conference, OCSC 2007, Held as Part of HCI International 2007, Beijing, China, July 22-27, 2007. Proceedings 2*, pages 445–454. Springer, 2007.
- [658] Robert A Virzi, Jeffrey L Sokolov, and Demetrios Karis. Usability problem identification using both low-and high-fidelity prototypes. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 236–243, 1996.
- [659] Miriam Walker, Leila Takayama, and James A Landay. High-fidelity or low-fidelity, paper or computer? choosing attributes when testing web prototypes. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 46, pages 661–665. Sage Publications Sage CA: Los Angeles, CA, 2002.
- [660] Matthew E. Walsh and Gigi Kwick Gronvall. Virologist opinions: An important component for the governance of the convergence of artificial intelligence and dual-use research of concern. *Applied Biosafety: Journal of the American Biological Safety Association*, 30:124–131, 2025.
- [661] April Yi Wang, Anant Mittal, Christopher A. Brooks, and Steve Oney. How data scientists use computational notebooks for real-time collaboration. *Proceedings of the ACM on Human-Computer Interaction*, 3:1 – 30, 2019.
- [662] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi (Jim) Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *ArXiv*, abs/2305.16291, 2023.

- [663] Leijie Wang, Nicholas Vincent, Julija Rukanskaitė, and Amy Xian Zhang. Pika: Empowering non-programmers to author executable governance policies in online communities. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2024.
- [664] Qiaosi Wang, Michael Madaio, Shaun Kane, Shivani Kapania, Michael Terry, and Lauren Wilcox. Designing responsible ai: Adaptations of ux practice to meet responsible ai challenges. CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.
- [665] Rose E Wang and Dorottya Demszky. Is chatgpt a good teacher coach? measuring zero-shot performance for scoring and providing actionable insights on classroom instruction. *arXiv preprint arXiv:2306.03090*, 2023.
- [666] Rose E Wang, Ana T Ribeiro, Carly D Robinson, Susanna Loeb, and Dora Demszky. Tutor copilot: A human-ai approach for scaling real-time expertise. *arXiv preprint arXiv:2410.03017*, 2024.
- [667] Ruotong Wang, Xinyi Zhou, Lin Qiu, Joseph Chee Chang, Jonathan Bragg, and Amy X Zhang. Social-rag: Retrieving from group interactions to socially ground proactive ai generation to group preferences. *arXiv preprint arXiv:2411.02353*, 2024.
- [668] Ruoyao Wang, Peter Alexander Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Conference on Empirical Methods in Natural Language Processing*, 2022.
- [669] Xingbo Wang, Samantha Lee Huey, Rui Sheng, Saurabh Mehta, and Fei Wang. Scidasynth: Interactive structured knowledge extraction and synthesis from scientific literature with large language model. *ArXiv*, abs/2404.13765, 2024.
- [670] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.
- [671] Zijie J Wang, Aishwarya Chakravarthy, David Munechika, and Duen Horng Chau. Wordflow: Social prompt engineering for large language models. *arXiv preprint arXiv:2401.14447*, 2024.
- [672] Zijie J Wang, Chinmay Kulkarni, Lauren Wilcox, Michael Terry, and Michael Madaio. Farsight: Fostering responsible ai awareness during ai application prototyping. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–40, 2024.

- [673] Zijie J Wang, Chudi Zhong, Rui Xin, Takuya Takagi, Zhi Chen, Duen Horng Chau, Cynthia Rudin, and Margo Seltzer. Timbertrek: Exploring and curating sparse decision trees with interactive visualization. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pages 60–64. IEEE, 2022.
- [674] Anne L Washington. How to argue with an algorithm: Lessons from the compas-publica debate. *Colo. Tech. LJ*, 17:131, 2018.
- [675] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *arXiv preprint arXiv:2307.02483*, 2023.
- [676] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- [677] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.
- [678] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [679] Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229, 2022.
- [680] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.
- [681] Yotam Wolf, Noam Wies, Yoav Levine, and Amnon Shashua. Fundamental limitations of alignment in large language models. *arXiv preprint arXiv:2304.11082*, 2023.
- [682] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics*, 22(1):649–658, 2015.
- [683] Michael Wooldridge. *An introduction to multiagent systems*. John wiley & sons, 2009.
- [684] Scott Wu. Introducing Devin, the first AI software engineer. <https://www.cognition.ai/blog/introducing-devin>, 2024.

- [685] Scott Wu. Introducing Devin, the first AI software engineer. <https://www.cognition.ai/blog/introducing-devin>, 2024.
- [686] Sherry Wu, Hua Shen, Daniel S Weld, Jeffrey Heer, and Marco Tulio Ribeiro. Scatter-shot: Interactive in-context example curation for text transformation. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 353–367, 2023.
- [687] Tongshuang Wu, Ellen Jiang, Aaron Donsbach, Jeff Gray, Alejandra Molina, Michael Terry, and Carrie J Cai. Promptchainer: Chaining large language model prompts through visual programming. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–10, 2022.
- [688] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–22, 2022.
- [689] Tongshuang Sherry Wu, Michael Terry, and Carrie J. Cai. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022.
- [690] Tongshuang Sherry Wu, Haiyi Zhu, Maya Albayrak, Alexis Axon, Amanda Bertsch, Wenxing Deng, Ziqi Ding, Bill Boyuan Guo, Sireesh Gururaja, Tzu-Sheng Kuo, Jenny T Liang, Ryan Liu, Ihita Mandal, Jeremiah Milbauer, Xiaolin Ni, N. Padmanabhan, Subhashini Ramkumar, Alexis Sudjianto, Jordan Taylor, Ying-Jui Tseng, Patricia Vaidos, Zhijin Wu, Wei Wu, and Chenyang Yang. Llms as workers in human-computational algorithms? replicating crowdsourcing pipelines with llms. *ArXiv*, abs/2307.10168, 2023.
- [691] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381, 2022.
- [692] Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*, 2025.
- [693] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Qin Liu, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huan, and Tao Gui. The rise

- and potential of large language model based agents: A survey. *ArXiv*, abs/2309.07864, 2023.
- [694] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Jing Hua Toh, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.
- [695] Tianbao Xie, Fan Zhou, Zhoujun Cheng, Peng Shi, Luoxuan Weng, Yitao Liu, Toh Jing Hua, Junning Zhao, Qian Liu, Che Liu, et al. Openagents: An open platform for language agents in the wild. *arXiv preprint arXiv:2310.10634*, 2023.
- [696] Frank F Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.
- [697] John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793*, 2024.
- [698] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *ArXiv*, abs/2405.15793, 2024.
- [699] Qian Yang. Machine learning as a ux design material: how can we imagine beyond automation, recommenders, and reminders? In *AAAI Spring Symposium*, 2018.
- [700] Qian Yang, Justin Cranshaw, Saleema Amershi, Shamsi T Iqbal, and Jaime Teevan. Sketching nlp: A case study of exploring the right things to design with language intelligence. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.
- [701] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. Re-examining whether, why, and how human-ai interaction is uniquely difficult to design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI ’20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.
- [702] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. Re-examining whether, why, and how human-ai interaction is uniquely difficult to design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

- [703] Stephen Yang, Jonathan Dorthheimer, Aaron Sprecher, and Qian Yang. When design workshops meet chatbots: Meaningful participation at scale? *International Journal of Architectural Computing*, 22(2):216–237, 2024.
- [704] Bingsheng Yao, Chaoran Chen, April Yi Wang, Sherry Tongshuang Wu, Toby Jia-jun Li, and Dakuo Wang. From human-human collaboration to human-agent collaboration: A vision, design philosophy, and an empirical framework for achieving successful partnerships between humans and llm agents. *arXiv preprint arXiv:2602.05987*, 2026.
- [705] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.
- [706] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv*, abs/2305.10601, 2023.
- [707] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *ArXiv*, abs/2210.03629, 2022.
- [708] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.
- [709] Andre Ye, Jared Moore, Rose Novick, and Amy X Zhang. Language models as critical thinking tools: A case study of philosophers. *arXiv preprint arXiv:2404.04516*, 2024.
- [710] Ryan Yen, Jiawen Stefanie Zhu, Sangho Suh, Haijun Xia, and Jian Zhao. Coladder: Manipulating code generation via multi-level blocks. In *ACM Symposium on User Interface Software and Technology*, 2024.
- [711] Nur Yildirim, Alex Kass, Teresa Tung, Connor Upton, Donnacha Costello, Robert Giusti, Sinem Lacin, Sara Lovic, James M O’Neill, Rudi O’Reilly Meehan, et al. How experienced designers of enterprise applications engage ai as a design material. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2022.
- [712] Nur Yildirim, Mahima Pushkarna, Nitesh Goyal, Martin Wattenberg, and Fernanda Viégas. Investigating how practitioners use human-ai guidelines: A case study on the people + ai guidebook. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI ’23, New York, NY, USA, 2023. Association for Computing Machinery.

- [713] Junnan Yu. Participatory design revisited: framings, key features, and its boundary with co-design. *CoDesign*, pages 1–30, 2025.
- [714] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. Wordcraft: Story writing with large language models. In *27th International Conference on Intelligent User Interfaces*, IUI '22, page 841–852, New York, NY, USA, 2022. Association for Computing Machinery.
- [715] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.
- [716] Matei Zaharia, Omar Khattab, Lingjiao Chen, Jared Quincy Davis, Heather Miller, Chris Potts, James Zou, Michael Carbin, Jonathan Frankle, Naveen Rao, and Ali Ghodsi. The shift from models to compound ai systems. <https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/>, 2024.
- [717] Øivind Klungseth Zahlse, Elena Parmiggiani, and Yngve Dahl. Challenges of scaling participatory design: A systematic literature review. In *Proceedings of the 34th Australian Conference on Human-Computer Interaction*, pages 143–159, 2022.
- [718] Theodore Zamenopoulos and Katerina Alexiou. *Co-design as collaborative research*. Bristol University/AHRC Connected Communities Programme, 2018.
- [719] J Diego Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–21, 2023.
- [720] Hongli Zhan, Allen Zheng, Yoon Kyung Lee, Jina Suh, Junyi Jessy Li, and Desmond Ong. Large language models are capable of offering cognitive reappraisal, if guided. In *First Conference on Language Modeling*, 2024.
- [721] Alice Qian Zhang, Ryland Shaw, Jacy Reese Anthis, Ashlee Milton, Emily Tseng, Jina Suh, Lama Ahmad, Ram Shankar Siva Kumar, Julian Posada, Benjamin Shestakofsky, et al. The human factor in ai red teaming: Perspectives from social and collaborative computing. In *Companion Publication of the 2024 Conference on Computer-Supported Cooperative Work and Social Computing*, pages 712–715, 2024.
- [722] Alice Qian Zhang, Jiayin Zhi, Srravya Chandhramowuli, Hong Shen, Laura Dabbish, Theodora Skeadas, Sarah Amos, and Jina Suh. The work of ai red teaming: Automation and the human infrastructure. In *Companion Publication of the 2025 Conference on Computer-Supported Cooperative Work and Social Computing*, pages 84–87, 2025.

- [723] Amy X Zhang, Grant Hugh, and Michael S Bernstein. Policykit: building governance in online communities. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 365–378, 2020.
- [724] Haoqi Zhang. Searching for the non-consequential: Dialectical activities in hci and the limits of computers. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery.
- [725] Jifan Zhang, Henry Sleight, Andi Peng, John Schulman, and Esin Durmus. Stress-testing model specs reveals character differences among language models. *arXiv preprint arXiv:2510.07686*, 2025.
- [726] Xingjian Zhang, Yutong Xie, Jin Huang, Jinge Ma, Zhaoying Pan, Qijia Liu, Ziyang Xiong, Tolga Ergen, Dongsub Shim, Honglak Lee, and Qiaozhu Mei. Massw: A new dataset and benchmark tasks for ai-assisted scientific workflows. 2024.
- [727] Zheng Zhang, Jie Gao, Ranjodh Singh Dhaliwal, and Toby Jia-Jun Li. Visar: A human-ai argumentative writing assistant with visual programming and rapid draft prototyping. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–30, 2023.
- [728] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR, 2021.
- [729] Chengbo Zheng, Dakuo Wang, April Yi Wang, and Xiaojuan Ma. Telling stories from computational notebooks: Ai-assisted presentation slides creation for presenting data science work. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022.
- [730] Qingxiao Zheng, Minrui Chen, Hyanghee Park, Zhongwei Xu, and Yun Huang. Evaluating non-ai experts’ interaction with ai: A case study in library context. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2025.
- [731] Andy Zhou, Kevin Wu, Francesco Pinto, Zhaorun Chen, Yi Zeng, Yu Yang, Shuang Yang, Sanmi Koyejo, James Zou, and Bo Li. Autoredteamer: Autonomous red teaming with lifelong attack integration. *arXiv preprint arXiv:2503.15754*, 2025.
- [732] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *ArXiv*, abs/2310.04406, 2023.

- [733] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.
- [734] Xuhui Zhou, Hyunwoo Kim, Faeze Brahman, Liwei Jiang, Hao Zhu, Ximing Lu, Frank Xu, Bill Yuchen Lin, Yejin Choi, Niloofar Miresghallah, et al. Haicosystem: An ecosystem for sandboxing safety risks in human-ai interactions. *arXiv preprint arXiv:2409.16427*, 2024.
- [735] Yuchen Zhuang, Xiang Chen, Tong Yu, Saayan Mitra, Victor S. Bursztyn, Ryan A. Rossi, Somdeb Sarkhel, and Chao Zhang. Toolchain*: Efficient action space navigation in large language models with a* search. *ArXiv*, abs/2310.13227, 2023.
- [736] Jonathan Zittrain. Three eras of digital governance. 2019.