Towards Multimodal Interactive Intelligence

Yushi Hu

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

Doctor of Philosophy

University of Washington
2025

Reading Committee:
Mari Ostendorf, Co-Chair
Noah A. Smith, Co-Chair
Luke Zettlemoyer
Hannaneh Hajishirzi

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Towards Multimodal Interactive Intelligence

Yushi Hu

Co-chairs of the Supervisory Committee:

Professor Mari Ostendorf
Electrical and Computer Engineering

Professor Noah A. Smith
Computer Science and Engineering

Great progress has been made in multimodal generative AI models. However, these models still have limitations. For example, they struggle when handling multimodal data and multi-turn interactions. One reason is the lack of training data for these problems. The research community lacks multimodal data on the scale of single modality data and lacks lengthy multi-turn interaction data. Additionally, there are not many well-defined tasks for these problems, preventing researchers from understanding models' performance and leaving them without meaningful optimization goals. In this thesis, we work toward building better multimodal intelligence. We focus on three types of abilities: multimodal understanding, multimodal generation, and grounded multi-turn interactions. For each aspect, we explore the limitations of current models, proposing new tasks and evaluation methods for capabilities that remain beyond the reach of existing models.

Identifying these weaknesses, we introduce novel methods and evaluations for multimodal interactive intelligence to address these challenges. This approach enhances existing AI models through AI-AI interactions and human-AI interactions, enabling collaboration across modalities.

For multimodal understanding models, we propose BLINK, a benchmark that focuses on core visual perception abilities not found in other evaluations. Most BLINK tasks can be solved by humans in a "blink"

of the eye, but pose significant challenges for the latest multimodal language models (LMs). To address this weakness, we propose Visual Sketchpad, a framework that allows models to think step-by-step across modalities. This framework empowers LMs to have more diverse interactions, for example, with vision expert models. Such interactions compensate for what existing models miss and greatly enhance models' multimodal understanding abilities.

For image generation models, we tackle the long-standing problem that these models do not effectively follow text instructions. We propose TIFA, which uses multimodal LMs to evaluate generated images, providing an efficient evaluation metric that aligns well with human judgment. Moreover, we show that TIFA can work as an effective training signal to improve text-image alignment in image generation.

Finally, we focus on grounded dialogue systems. We provide a framework that allows AI agents to be evaluated with either simulated or real users, using end-to-end dialogue-level objectives. To demonstrate the use of this framework, we introduce NavigationBench, a novel task that simulates dialogues between a user and a virtual navigation assistant in a car. It also features a simulated user trained with the latest LM technologies, allowing researchers to simulate multi-turn dialogues for automatic dialogue-level comparisons of AI assistants. Using this framework, we study the performance and verbosity of different agent LLMs.

Acknowledgements

I would like to express my deepest gratitude to my exceptional advisors, Prof. Mari Ostendorf and Prof. Noah A. Smith. Their unwavering support, sharp insights, and generous mentorship have shaped every step of my PhD journey. From cultivating my appreciation for thoughtful, impactful research to sharpening my critical thinking and communication skills, they have profoundly influenced my development as a researcher. I am especially thankful for the trust and freedom they gave me from day one — always treating me with respect, compassion, and encouragement. They have made me more open to new ideas, more patient in the face of challenges, and a better storyteller. I could not have asked for better mentors.

Although Prof. Ranjay Krishna was not formally on my committee, he played a pivotal role in my journey. As a co-author on every one of my vision-related projects, he has been a constant collaborator, mentor, and friend. Our synergy has led to some of the most exciting and impactful work in my PhD. Ranjay has always offered me his full support and time, unconditionally. He has been a de facto third advisor, and I am deeply grateful for everything I have learned from working with him.

I am also fortunate to have had Prof. Luke Zettlemoyer, Prof. Hannaneh Hajishirzi, and Prof. Aylin Caliskan serve on my committee. I deeply appreciate their thoughtful feedback and guidance, which have meaningfully improved this thesis. I have had the pleasure of collaborating with Luke and Hanna on several projects — both are brilliant researchers whose insights and vision have elevated our work and inspired me to think more broadly.

My sincere thanks go to all of my research collaborators over the years, especially Weijia Shi, Zeqiu Wu, Xingyu Fu, Wei-Chiu Ma, and Benlin Liu, whose contributions to the work presented in this thesis were invaluable. Many of my ideas have been sparked during our conversations and brainstorms, and this body of work would not exist without them. I also thank my internship mentors — Kunpeng Li, Xiaoliang Dai, and Peizhao Zhang at Meta, Otilia Stretcu at Google, and Chunlei Zhang at Tencent — for their support and guidance during my industry experiences, which broadened my perspective and enriched my research.

I am proud to be a part of the vibrant research communities at the TIAL Lab, Noah's Ark, RAIVN Group, and UWNLP. These groups have provided a supportive, intellectually stimulating environment, and many of their members have become close collaborators and friends. I'm grateful to everyone I've had the chance to work and grow alongside.

To my parents and family, thank you for your unwavering love and support over the years. Your values, encouragement, and belief in me have been the foundation on which I've built my life. You've taught me resilience, kindness, and the courage to pursue what I love. I am also deeply grateful to my friends — both near and far — for the joy, laughter, and grounding moments we've shared. Finally, I am endlessly thankful to my partner, Zhiying Zhu, whose constant love, care, and companionship have been my anchor. Thank you for walking with me through every high and low, and for always believing in me.

DEDICATION

To my grandfather, *Chengxing Hu* (born in 1945 in Harbin, passed away in 2024 in Beijing), who raised me to be a kind person. His presence shaped who I am, and his love continues to guide me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Current generative AI models are typically large, monolithic models trained with vast amounts of data and computational resources [Brown et al., 2020; OpenAI, 2023b; Anthropic, 2024b; Touvron et al., 2023; Dubey et al., 2024; Team et al., 2023; Deitke et al., 2024; Podell et al., 2023]. As AI is applied to increasingly complex challenges, these models reveal certain limitations. For example, even top proprietary models struggle with multimodal tasks [Fu et al., 2024; Yue et al., 2024; Hu et al., 2024b] and with tasks involving extended multi-turn interactions [Wang et al., 2024b; Hu et al., 2022]. A primary reason for these limitations is the lack of suitable training data: compared to the vast volume of text data available, multimodal data remains relatively scarce [Deitke et al., 2024]. Additionally, gathering data that captures multi-party, multi-turn interactions is particularly challenging [Lu et al.]. Moreover, automatically evaluating these complex tasks, such as image generation and multi-turn human-AI interaction, is itself difficult. Having automatic evaluations that correlate well with humans has been helpful in expediting the research process in areas like speech recognition and machine translation. Nevertheless, for these complex tasks, developing reliable benchmarks and evaluation metrics remains an open and pressing challenge.

This thesis aims to push the boundaries of AI models further. We first investigate their limitations and develop novel evaluations for capabilities that extend beyond the reach of existing models. To address these challenges, we draw inspiration from recent work on augmenting models with specialized modules. For instance, researchers have long recognized that integrating retrieval modules with large language models (LLMs) can enhance factual accuracy and utility, and have used LLMs as components of powerful agents that interact with their environment via reasoning, planning, and tool use [Yao et al., 2023]. Building upon these foundations, we argue that AI should not be constrained to single-modal monolithic systems that merely produce an output given an input. We introduce the concept of **multimodal interactive intelligence**, which we define as an AI paradigm where systems engage in meaningful multi-turn interactions, collaborate across multiple modalities, and learn through interactions. To achieve this goal, we need to develop a broad range of new tasks and evaluation methods, interaction paradigms, and learning algorithms. Our ultimate goal is to build AI systems that demonstrate above capabilities in diverse and complex real-world scenarios. In the rest of the introduction, we outline our solutions toward this goal for three types of model abilities: multimodal understanding, multimodal generation, and grounded multi-turn interactions.

1.1 Enhance Multimodal Understanding with Interactions

Perception is the basis for multimodal intelligence to understand and reason about the world. In our work BLINK [Fu et al., 2024], we identify a critical gap in current multimodal LLMs: while these models can process visual inputs, they lack fundamental perception abilities that humans can perform "within a blink of the eye." Through comprehensive evaluation across 14 carefully designed visual perception tasks—including relative depth estimation, visual correspondence, and multi-view reasoning—we discover that even state-of-the-art multimodal models like GPT-4V achieve only 51.26% accuracy, dramatically underperforming humans (95.70%) and often barely outperforming random chance. This finding reveals that multimodal models can "see but not perceive"—they process visual inputs at a superficial level but struggle with tasks that require nuanced visual understanding. Our analysis also demonstrates that specialist computer vision models significantly outperform general multimodal LLMs on these perception tasks, suggesting potential pathways for improvement.

Building on these insights, we introduce Visual Sketchpad [Hu et al., 2024a], a framework that equips multimodal language models with the ability to reason step-by-step across modalities. Inspired by how humans draw to facilitate reasoning—creating auxiliary lines when solving geometry problems, marking when analyzing images, and circling on maps—Sketchpad enables models to produce visual artifacts with programs and models as part of the multimodal reasoning process. This approach significantly improves performance across diverse mathematical and visual reasoning tasks, with an average absolute gain of 12.7%

accuracy on math tasks and 8.6% on vision tasks over GPT-4o. Specifically, Sketchpad allows models to leverage specialist vision models during the sketching process, including drawing bounding boxes with object detection models, generating depth maps, and segmenting images with masks, which substantially enhances visual perception capabilities. Our experiments demonstrate that GPT-4o with Sketchpad sets new state-of-the-art results on multiple benchmarks, highlighting the potential of interactive visual reasoning to overcome the perception limitations identified in BLINK.

1.2 Train and Evaluate Image Generation Models via Interaction

Beyond understanding, generation is also an important aspect for multimodal intelligence. A key challenge of generation is its faithfulness to users' inputs. To address this challenge, we introduce TIFA (Text-to-Image Faithfulness evaluation with question Answering) [Hu et al., 2023b], a reference-free evaluation metric that measures the faithfulness of generated images to their text prompts through visual question answering. Unlike existing metrics that rely on CLIP [Hessel et al., 2021] or image captioning, TIFA uses a language model to automatically generate multiple-choice questions about the text prompt, then evaluates whether a VQA model can correctly answer these questions using the generated image. We also introduce the TIFA benchmark, including 4,000+ diverse prompts and 25,000+ questions. Through extensive experiments on the benchmark, we demonstrate that TIFA has substantially higher correlation with human judgments (Spearman's $\rho = 0.60$) than previous metrics like CLIPScore (0.33) and caption-based approaches (0.34). Our analysis reveals that current text-to-image models excel at rendering common objects, animals, and colors, but struggle with counting, spatial relations, and composing multiple objects. TIFA's design allows for fine-grained analysis across twelve different categories of visual elements, providing interpretable feedback that can guide future improvements in text-to-image generation systems. The TIFA benchmark serves as a standardized testbed that researchers can use to compare models fairly and identify specific areas where current models need improvement. TIFA has been widely adopted in recent years. Follow-up studies show that TIFA also works well as a training signal. For example, our work Dreamsync [Sun et al., 2025] shows that TIFA, together with an aesthetic reward, can greatly improve image generation quality through a few iterations of rejection sampling training.

1.3 Towards AI Assistants in Grounded Multi-turn Interactions

To move beyond simplistic input-output interactions and create AI systems that function effectively in real-world scenarios, we develop NavigationBench, a novel evaluation framework for assessing AI assistants in simulated navigation contexts. This benchmark simulates dynamic dialogues between users and navigation assistants where users need to make unplanned stops during their journeys, necessitating collaborative problem-solving through multi-turn interactions. Unlike traditional benchmarks that rely on static prompt-response pairs, NavigationBench creates a fully grounded environment with functional map capabilities, realistic scenarios collected from human annotators, and a fine-tuned open-source LLM that simulates user behavior. The framework evaluates assistant performance through human-annotated rankings of potential solutions and captures critical efficiency metrics such as token usage and conversation length. Through comprehensive studies with diverse language models and interaction protocols, NavigationBench offers valuable insights into building AI assistants that can effectively collaborate with users in complex, real-world scenarios while maintaining computational efficiency.

1.4 Thesis Overview

This thesis advances multimodal interactive intelligence by developing novel benchmarks, evaluation methods, and interaction paradigms that enable AI systems to better understand, generate, and interact across multiple modalities in real-world scenarios. The remainder of this thesis is organized as follows. In Chapter 2, we provide background on multimodal language models, media generation models, evaluations, and interactive AI systems, establishing the foundation for our contributions. Chapter 3 introduces BLINK, a comprehensive benchmark that reveals fundamental perception gaps in current multimodal LLMs. Chapter 4 presents Visual Sketchpad, a framework that enables multimodal LLMs to generate intermediate visual artifacts during reasoning, significantly enhancing performance on mathematical and visual tasks. Chapter 5 details TIFA, a novel metric for evaluating text-to-image generation faithfulness using multimodal language models, providing fine-grained analysis that guides future model development. Chapter 6 presents NavigationBench, a framework for evaluating AI assistants in realistic multi-turn interactions within a grounded map environment. We conclude in Chapter 7 by summarizing our contributions and outlining promising directions for further research in multimodal interactive intelligence.

Chapter 2

Background

In this chapter, we first provide a review of background work. In §2.1, we briefly review recent works on multimodal language models, discuss evaluations on their visual understanding abilities, and on augmenting them with tools and agent frameworks. This section laid the basis for BLINK [Fu et al., 2024] and Visual Sketchpad [Hu et al., 2024a]. In §2.2, we discuss existing image generation models and evaluation metrics, which are the basis for TIFA [Hu et al., 2023b]. Finally, in §2.3, we provides an overview of prior works using language models to interact with humans and with the world, which is closely related to NavigationBench and Visual Sketchpad [Hu et al., 2024a].

2.1 Multimodal Language Models

Inspired by the impressive success in recent large language models [Brown et al., 2020; OpenAI, 2023b; Touvron et al., 2023; Zheng et al., 2023; Chowdhery et al., 2023], a sequence of studies explore multimodal LMs that can jointly understand vision and language information and generate textual answers through adding a modality adaption structure between a frozen visual encoder [Radford et al., 2021; Sun et al., 2023; Fang et al., 2023] and a frozen LLM [Touvron et al., 2023; Zheng et al., 2023]. Flamingo [Alayrac et al., 2022] and BLIP-2 [Li et al., 2023b] are two of the earliest works to explore these transformer-based multi-modality conjunction structures. They first pre-train on image-text pair datasets [Lin et al., 2014; Krishna et al., 2017; Schuhmann et al., 2021; Changpinyo et al., 2021] and then fine-tune on task-specific datasets such as visual question answering (VQA) [Antol et al., 2015; Goyal et al., 2017]. Starting from LLaVA [Liu et al., 2024a,b], people use LLM-synthesized instruction-following chat data (which are in VQA format) for instruction tuning and achieve much better results on vision-language tasks like dense image captioning and VQA [Dai et al., 2023a; Chen et al., 2023a; Team et al., 2023; Bai et al., 2023]. There have been extended studies that explore further capabilities of multimodal LLMs, especially on VQA reasoning [Zellers et al., 2019; Fu et al., 2023c, 2022; Schwenk et al., 2022; Fu et al., 2023d; Yan et al.]. However, they mainly focus on the textual reasoning abilities [Wei et al., 2022] within the multimodal LMs and do not emphasize visual perception. The work in this thesis, BLINK [Fu et al., 2024] and Visual Sketchpad [Hu et al., 2024a], further expands the capability boundaries of multimodal LMs. BLINK highlights crucial aspects of visual perception that have been overlooked when evaluating multimodal LMs. We discuss more details on prior multimodal LM evaluations in §2.1.1. Visual Sketchpad augments multimodal LMs with the ability to reason across modalities, enabling multimodal LMs to achieve new state-of-the-art results on multiple complex visual reasoning tasks. We provide a review of relevant LM tool-use studies in §2.1.2.

2.1.1 Evaluations for Multimodal Understanding

Traditional vision-language datasets are designed to assess single-task capabilities, such as optical character recognition (OCR) [Liu et al., 2024f], image captioning [Lin et al., 2014], and visual question answering [Antol et al., 2015; Goyal et al., 2017]. However, these datasets are often not comprehensive enough to holistically assess multimodal LMs on general perception and reasoning abilities. Many recent papers have built more comprehensive VQA benchmarks. MME [Fu et al., 2023a] is one of the earliest holistic benchmarks containing multi-modal Yes/No questions on the defined visual perception and language reasoning tasks. MM-Vet [Yu et al., 2024] includes six sub-features from the previous datasets, including recognition-focused questions, OCR, and math, providing a diverse evaluation set. MMBench [Liu et al., 2024e] covers more subjects in its visual questions, and proposes "circular evaluation," which permutes the order of the choices for each question to address the positional bias in multiple-choice questions. The Seed-Bench [Li et al., 2023a, 2024b] benchmark has a more diverse source of visual inputs, including multiple-image inputs and video, and further expands the number of subjects in the visual questions. However, the visual perception questions in MME, MMBench, MM-Vet, and Seed-Bench are mainly extracted from existing VQA datasets or generated by GPT [OpenAI, 2023b] from image descriptions such as COCO-Caption [Lin et al., 2014], and are recognition-focused, covering topics such as object (attribute) recognition, and OCR. Some other multimodal benchmarks have distinct focuses. MMMU [Yue et al., 2024] aims at achieving expert-level artificial general intelligence by collecting domain-knowledge-required questions. HallusionBench [Guan et al., 2024] mainly tests the language hallucination and visual illusion phenomena. MathVista [Lu et al., 2024b] presents exclusively mathematical domain visual questions based on images such as charts, tables, and diagrams. These benchmarks require only a limited range of human perception abilities and therefore cannot measure model visual perceptions holistically. To address this challenge, we propose BLINK [Fu et al., 2024], a benchmark consists of 14 classical computer vision tasks that focus on several core visual perception abilities that humans possess, and find that these questions are extremely challenging for existing multimodal LMs. We present this work in detail in §3.

2.1.2 LMs as Tool-use Agents

Recent work has started to treat LMs as agents that can both reason and act [Yao et al., 2023; Ma et al., 2024; Wu et al., 2023; Shi et al., 2024; Yasunaga et al., 2023]. Researchers have applied this idea to software engineering [Jimenez et al., 2024; Zhang et al., 2024a; Hong et al., 2024], robotics [Nasiriany et al., 2024], vision [Liu et al., 2024c; Yang et al., 2023c], and GUI navigation [Yan et al., 2023; Koh et al., 2024; Xie et al., 2024]. An important capability of LM agents is to use tools to facilitate problem solving [Schick et al., 2023]. Specifically, for multimodal problems, researchers have demonstrated the possibility of decomposing complex tasks into simpler substeps that can each be solved using vision tools [Yang et al., 2023c; Zeng et al., 2023; Hu et al., 2024b, 2023a]. Among them, the most recent ones are Visprog [Gupta and Kembhavi, 2023] and ViperGPT [Surís et al., 2023]. They use LMs to generate Python code, which sequencially invokes specialized vision tools. However, among all these works, the process of "reasoning" is text-only: all tools only output text, and the multimodal LMs only take these text inputs to plan the next steps. Our Visual Sketchpad [Hu et al., 2024a] addresses this limitation. Our framework enables the models to create visual artifacts to facilitate reasoning, which greatly boosts their multimodal reasoning abilities. We discuss this work in detail in §4.

2.2 Text-to-Image Generation Models

Recently, large-scale pre-trained diffusion models [Ho et al., 2020; Rombach et al., 2021; Saharia et al., 2022; Dai et al., 2023b; Podell et al., 2023; Esser et al., 2024] have demonstrated remarkable capabilities in text-to-image generation. These models typically employ an architecture where the text encoder first encodes the input prompt into a text embedding, and then feeds the embedding into the diffusion backbone via cross-attention. The diffusion model subsequently generates VAE latent vectors, which are decoded into an image. A core challenge of image generation is model evaluation. Prior work usually relies on humans to evaluation the model-generated images. Since human annotators can be subjective, these studies usually ask annotators to perform pairwise comparisons of images generated by different models. This approach can be prohibitively expensive and hard to scale. Also, it does not provide an absolute score that can be compared across different models.

Designing automatic evaluation metrics of the quality of machine-generated images has always been a major challenge in computer vision. There are two aspects to evaluate, namely image quality and image-text alignment. Inception Score [Salimans et al., 2016] and FID [Heusel et al., 2017] are the most widely adopted metrics for image quality. They compare the features of the generated images and gold images extracted from a pre-trained Inception-V3 model [Szegedy et al., 2015] to evaluate the fidelity and diversity of generated images. However, they rely on ground-truth images and are based on a classification model, which makes them not suitable for complex datasets [Frolov et al., 2021]. For image-text alignment, prior metrics are mainly based on CLIP [Radford et al., 2021], captioning, and object detection models. CLIPScore [Hessel et al., 2021] and CLIP-R [Park et al., 2021] are based on the cosine similarity of image and text CLIP [Radford et al., 2021] embeddings. Other works first convert the images using a captioning model, and then compare the image caption with the text using metrics like CIDEr [Vedantam et al., 2015] and SPICE [Anderson et al., 2016]. A common problem for these metrics is their low correlation with human annotators [Hessel et al., 2021]. We introduce TIFA [Hu et al., 2023b] to address this challenge. Empowered by multimodal LMs, TIFA has a much better correlation with human annotators, and can provide fine-grained and interpretable analysis on different aspects of image generation. TIFA has been widely adopted in follow-up works and has become the basis of many latest image generation benchmarks [Cho et al.; Wiles et al., 2024; Li et al., 2024a; Lin et al., 2024]. We discuss this work in §5.

2.3 Interactive AI Systems

The field of interactive AI systems has witnessed significant advancements, particularly with the emergence of Large Language Models (LLMs). These models have demonstrated remarkable capabilities in collaborating with humans across a spectrum of tasks. For instance, prior work has leveraged human-LLM collaboration to synthesize NLI data [Liu et al., 2022a] and task-oriented dialogue [Lu et al.], to facilitate code generation [Yang et al., 2024a], and to enhance creative writing [Lee et al., 2022]. These examples underscore the broad potential of LLMs to augment human intelligence and creativity across various domains.

2.3.1 Task-Oriented Dialogue Systems

Task-oriented dialogue systems (TODs) constitute a specific category of interactive AI designed to assist users in accomplishing particular tasks through natural language conversations. These systems are engineered for efficiency in task completion within well-defined domains, finding practical applications in areas such as booking systems, customer service platforms, and information retrieval interfaces. The primary objective of a TOD is to guide the interaction toward a successful outcome, such as completing a reservation or answering a specific query. Most task-oriented dialogue systems operate with a backend containing structured data.

The evolution of TODs spans several decades. The Airline Travel Information System (ATIS) corpus pioneered this field by framing flight-booking as a language-understanding benchmark, remaining a classic dataset for intent/slot modeling [Hemphill et al., 1990]. A decade later, the CMU Let's Go bus-information system demonstrated that research prototypes could serve real commuters via telephone, highlighting the importance of robustness to noisy user speech [Raux et al., 2005]. These early efforts led to the Dialogue State Tracking Challenge (DSTC) series, which released shared human-machine dialogue corpora and evaluated systems on dialogue state tracking (DST) tasks [Williams et al., 2013].

With the advent of neural dialogue systems, researchers shifted toward richer, multi-domain settings. MultiWOZ introduced over 10,000 multi-turn conversations spanning seven domains, establishing itself as the de facto benchmark for modern TOD models [Budzianowski et al., 2018]. Concurrently, conversational question-answering tasks such as QuAC and CoQA blurred the boundary between TODs and reading comprehension, expanding dialogue systems to general information-seeking applications [Choi et al., 2018; Reddy et al., 2019].

Recently, with the emergence of more powerful language models, researchers have begun exploring LM interactions with complex environments beyond structured databases. For example, Lee et al. [2023b] assess LMs on human-AI collaborative tasks including crossword puzzles, summarization, and metaphor generation, while Wang et al. [2024b] measures LM performance when incorporating language feedback for code generation, decision-making, and mathematical problem-solving. However, this line of research predominantly focuses on relatively simple interactions where humans provide feedback to LLMs.

In this thesis, we introduce NavigationBench, a multi-turn dialogue task grounded in a map environment. This benchmark features a simulated user powered by state-of-the-art LLMs, enabling researchers to investigate more complex human-AI interactions. Additionally, by providing a simulated map environment, NavigationBench facilitates more sophisticated interactions between AI systems and virtual environments.

2.3.2 Evaluation of Dialogue Systems

The evaluation of dialogue systems represents a critical dimension of their development process, offering valuable insights into their performance, effectiveness, and the quality of user experience they provide. However, assessing dialogue quality remains inherently complex, particularly for open-domain systems, due to the subjective nature of conversation and the challenge of establishing objective metrics that consistently align with human judgments.

At the turn level, Dialogue State Tracking (DST) serves as a primary evaluation metric for goal-oriented dialogue systems. DST measures the accuracy with which a system tracks user intentions and the evolving dialogue state, typically represented as slot-value pairs. The Dialogue State Tracking Challenges (DSTC) have been instrumental in advancing this area by providing standardized datasets and evaluation frameworks [Williams et al., 2013]. Beyond DST, additional turn-level metrics assess response quality at each interaction, considering factors such as relevance to user input, coherence with preceding dialogue, linguistic fluency, and information content. These evaluations often involve measuring overlap with human-generated reference responses or employing trained models to predict the perceived quality of system outputs [Tao et al., 2018].

Evaluating the holistic effectiveness of dialogue systems requires consideration of system-level metrics that transcend individual turns. User satisfaction emerges as a paramount system-level metric, reflecting the comprehensive user experience and the system's capacity to fulfill user needs. The Amazon Alexa Prize, for instance, uses conversation length (number of turns) to gauge a system's ability to sustain user engagement [Venkatesh et al., 2018]. However, excessive conversational turns may result in unnecessary or irrelevant exchanges that correlate with negative user experiences.

In our NavigationBench, we evaluate dialogue systems using a task-level objective—specifically, the rank of the system-proposed stop. Unlike turn-level metrics or indirect measures such as conversation length, NavigationBench's evaluation metrics directly align with the dialogue's goal. Notably, leveraging recent advances in LLMs, our evaluation framework functions effectively with both human users and LLM-powered simulated users, enabling more efficient and rapid iteration in dialogue system development.

Chapter 3

BLINK: Multimodal Large Language Models Can See but Not Percieve

3.1 Introduction[1]

Compared to today, computer vision was originally attempting to interpret images as projections of 3D scenes, not just processing 2D arrays of flat "patterns" [Minsky and Papert, 1969; Marr, 2010]. In this pursuit, early research developed a series of intermediate tasks: they focused on understanding optical properties like reflectance [Wang and Adelson, 1993; Black and Anandan, 1993], 3D primitives through multi-view reasoning [Marr and Poggio, 1976; Hartley and Zisserman, 2003], geometric reasoning through depth estimation [Torralba and Oliva, 2002], instance recognition through visual correspondence [Lowe, 1999], affordance through keypoint grounding [Harris et al., 1988], and forensics through intrinsic images [Barrow et al., 1978]. Yet in the modern era of large language models (LLMs), the community has focused less on such perceptual tasks, and instead developed new tasks, mostly expressed in natural language, emphasizing the vision-language connection learned by multimodal LLMs [OpenAI, 2023b; Team et al., 2023; Chen et al., 2023c; Liu et al., 2023b, 2024b; Alayrac et al., 2022; Bai et al., 2023; Wang et al., 2024a; Dai et al., 2023a; Chen et al., 2023b; Dong et al., 2024; Lu et al., 2024a; Sarkar et al., 2024]. This might be because many

[1]This work involved several authors, and the primary authors are Xingyu Fu from the University of Pennsylvania and me. I proposed the idea and collected the data for visual correspondence, counting, localization, multi-view reasoning, spatial reasoning, relative depth, and relative reflectance tasks. Xingyu is in charge of other tasks. We jointly wrote the paper.

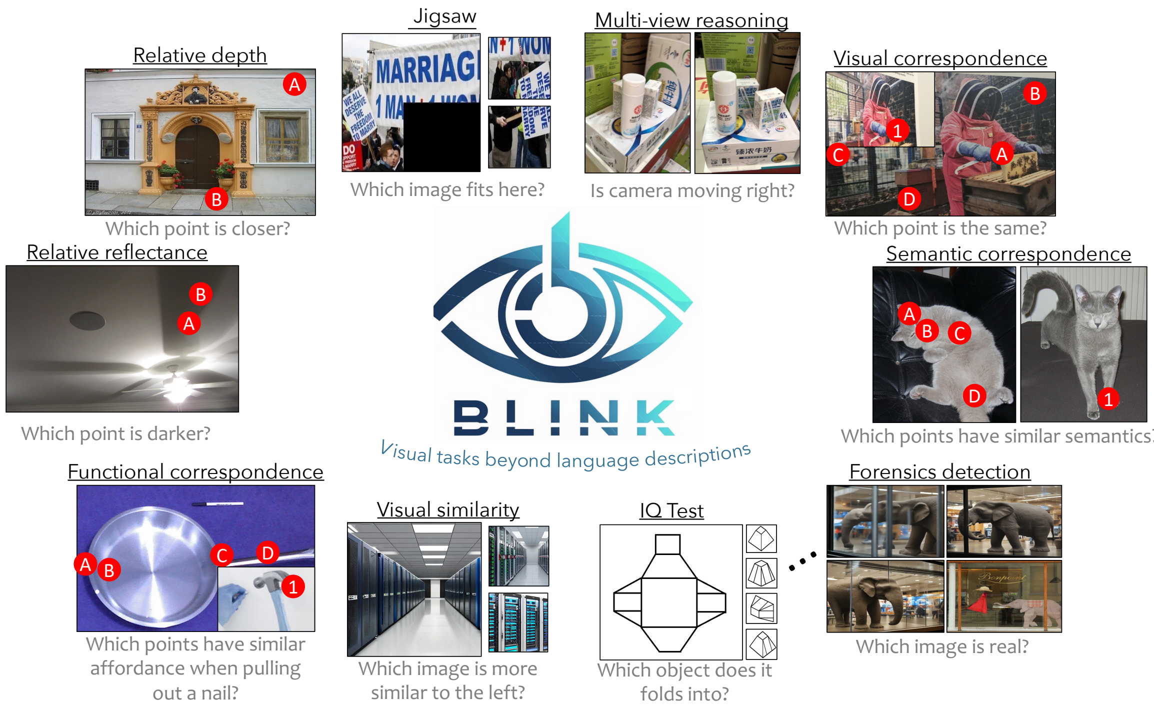


Figure 3.1: The BLINK Benchmark. BLINK contains 14 visual perception tasks that can be solved by humans "within a blink", but pose significant challenges for current multimodal LLMs. These tasks are inspired by classical computer vision problems and recast into multiple-choice questions for multimodal LLMs to answer. Notice that the visual prompts and questions in this figure are different from the actual ones used in the benchmark for illustrative purposes. The answers are as follows. Relative depth: B; jigsaw: A; multi-view reasoning: right; visual correspondence: A; semantic correspondence: C; forensics detection: final image; IQ test: D; visual similarity: upper one; functional correspondence: A; relative reflectance: they are about the same.

traditional computer vision tasks resist mediation through natural language, due to the inherent imprecision of language (*e.g.*, it is challenging to precisely pinpoint a spatial keypoint through language).

BLINK aims to highlight crucial aspects of visual perception that have been overlooked when evaluating multimodal LLMs. To appropriately position this work, let us revisit how we currently evaluate perception through using multimodal LLMs [Liu et al., 2024e; Li et al., 2023a, 2024b; Yue et al., 2024; Lu et al., 2024b; Liu et al., 2023a, 2024f]. While many of these benchmarks have been popularized as the de facto evaluation measures for influential models like GPT-4V and Gemini-Pro, they conflate perception with language knowledge and reasoning. At the risk of singling out one benchmark, let us consider two questions highlighted in the popular MMBench [Liu et al., 2024e]: "`<image 1> Why is this hummingbird called ruby-throated?`" and "`<image 2> What will happen next? A: the person is gonna`

Figure 3.2: Comparison between BLINK and previous benchmarks. BLINK has several novel features: (1) BLINK incorporates diverse visual prompts, like circles, boxes, and image masks, while previous benchmarks only have text questions and answers. (2) BLINK evaluates a more comprehensive range of visual perception abilities, like multi-view reasoning, depth estimation, and reflectance estimation. Prior benchmarks are generally more focused on recognition-based VQA. (3) BLINK contains "visual" commonsense problems that humans can answer within seconds, while prior benchmarks like [Yue et al., 2024] require domain knowledge. The samples of previous benchmarks are from [Liu et al., 2024e; Li et al., 2023a; Yue et al., 2024]. Part of our samples are curated from [Lai et al., 2021; Gupta et al., 2019; Fu et al., 2022; Ze and Wang, 2022; Chen et al., 2016; Bell et al., 2014; Fu et al., 2023b].

`laugh B: the person is gonna cry.`" For the first question, the vision subpart is to recognize the hummingbird. For the second, it only needs a coarse description of the image. Everything else is left to the language model to solve. Such a conflation has also been reported for other benchmarks by previous work [Yang et al., 2022; Hu et al., 2023a; Berrios et al., 2023]. Our experiments show that this conflation reductively evaluates perception as a dense captioning task. In other words, **by replacing the image with a task-agnostic dense caption, our experiments show that a "blind" GPT-4 performs well on these "multimodal tasks".**

In response, we propose BLINK. BLINK reimagines traditional computer vision problems through a format that allows us to evaluate multimodal LLMs. As partially demonstrated in Figure 3.1, BLINK consists of 14 classic computer vision tasks, ranging from low-level pattern matching (*e.g.*, visual correspondences estimation) to mid-level reasoning (*e.g.*, relative depth estimation), and extending to high-level visual understanding (*e.g.*, visual similarity). The image tasks are meticulously selected such that they are difficult to solve by reducing the evaluation using dense captioning; instead, the models must perceive the contents of the image(s) to answer. We recast each traditional task into a modern question-answering format, where answer choices are either images or text. BLINK contains 3.8K questions across 7.3K images, where questions may contain multiple images that are curated from a wide range of datasets [Lin et al., 2014; Krishna et al., 2017; Bell et al., 2014; Balntas et al., 2017; Chen et al., 2016; Gupta et al., 2019; Fu et al., 2023b, 2022], encompassing indoor household scenes as well as outdoor urban or natural environments. The questions and choices are either derived from the datasets, or manually written by humans. On average, each question can be solved by a human subject within a BLINK of an eye, except the IQ test.

We carefully evaluate 17 multimodal LLMs with various sizes (*i.e.*, 7B, 13B, 34B) on BLINK. We observe the paradox that **while these problems are easy for humans (**95.70% **average accuracy), they are extremely hard for existing multimodal LMs** – even GPT-4V model can only achieve 51.26% accuracy on average, which is 44.44% worse than humans, and 13.17% better than random guessing. We also experiment with specialist vision models and find that they perform much better than multimodal LLMs. For example, the specialist outperforms GPT-4V by 62.8% on visual correspondence estimation, 38.7% on relative depth estimation, and 34.6% on multi-view reasoning, in terms of absolute accuracy. Our findings indicate that the perceptual abilities of multimodal LLMs have been previously overestimated. Furthermore, these models may benefit from integrating insights from specialized models that excel in these areas. We believe BLINK can serve as an effective testbed for bridging the gap between traditional notions of perception and the modern generative capabilities of multimodal LLMs.

3.2 The BLINK Benchmark

Our goal is to faithfully evaluate the visual perception capabilities of existing Multimodal LLMs. We seek to study the visual perception gap between humans and machineries, and offer deeper insights into potential pathways towards achieving more generalized machine perception. Based on the observation that

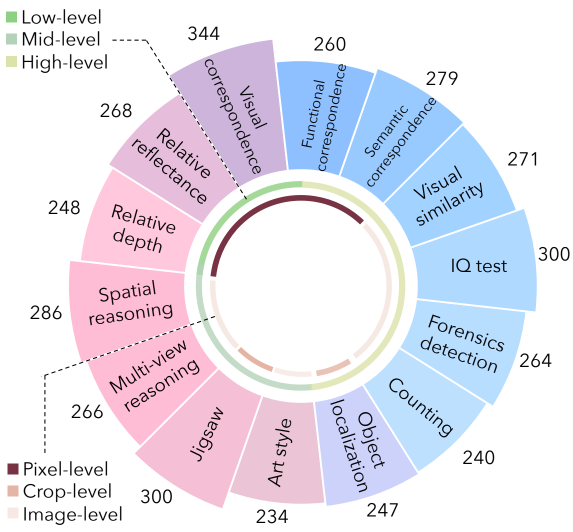


Figure 3.3: Statistics of BLINK. The benchmark includes 14 tasks, ranging from pixel-level to image-level perception, and low-level pattern matching (*e.g.*, visual correspondences estimation) to mid-level reasoning (*e.g.*, relative depth estimation), and extending to high-level visual understanding (*e.g.*, visual similarity).

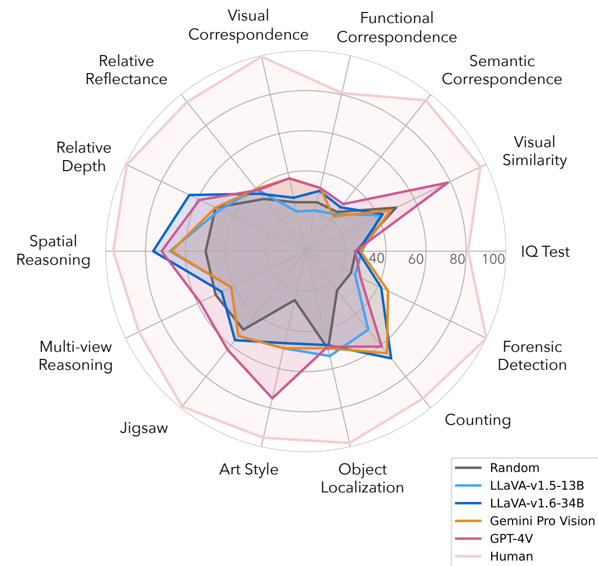


Figure 3.4: Accuracies of multimodal LLMs on BLINK test set. Please refer to Table 3.1 and §3.3.2 for more results and discussions.

existing benchmarks predominantly focus on evaluating visual recognition abilities, we introduce a novel benchmark, BLINK, designed to enable both quantitative and qualitative evaluation of the nuanced perception capabilities of multimodal LLM across various dimensions. We unfold this section by illustrating the overall design of BLINK (§3.2.1) and discussing its unique features comparing with previous benchmarks. Then we describe each task in detail, providing an in-depth explanation of the data curation process (§3.2.2).

3.2.1 Overview of BLINK

To ensure that one can effectively measure what Multimodal LLMs can or cannot perceive, we carefully select 14 tasks (see §3.2.2 for the full list) that are difficult to solve by reducing the evaluation into text-only questions using dense captioning. The tasks are drawn from either classic computer vision problems or recent applications of Multimodal LLMs, each of which requires a nuanced understanding of the visual data. They range from low-level pattern matching (*e.g.*, visual correspondence) to mid-level spatial reasoning (*e.g.*, relative depth), and up to high-level visual understanding (*e.g.*, visual similarity). This variety allows for a systematic exploration of Multimodal LLMs' capabilities across different perceptual complexity layers.

Furthermore, these visual tasks vary in granularity, ranging from pixels (*e.g.*, relative reflectance) to patches (*e.g.*, jigsaw) and extending to the full image (*e.g.*, forensic detection), enabling us to evaluate models' proficiency in observing at various scales.

To facilitate the evaluation of multimodal LLMs, we recast all tasks as multiple-choice question-answering problems. The options for answers may include images or texts, while the questions themselves can feature either single or multiple images. Prompts are designed to be both textual and visual in nature. We re-purposed several existing vision datasets as well as collected new data. In total, we contribute 3.9K multiple-choice questions and 7.3K images, with an even distribution between the validation and test sets. Numbers of each task are reported in Figure 3.3.

Key features of BLINK: Comparing with previous benchmarks, BLINK has the following novel features:

- **Visual prompting**: Unlike existing benchmarks that support only text prompting, BLINK features a variety of visual prompts. This enables one to highlight specific areas within images, facilitating the evaluation of Multimodal LLMs' detailed understanding of these regions. It also offers an interface for researchers to investigate the impact of visual prompting techniques.

- **Perception beyond recognition**: Besides visual recognition, BLINK considers a diverse set of visual perception abilities, such as 3D reasoning, geometric understanding, affordance reasoning, etc. The breadth allows one to evaluate Multimodal LLMs from an unique array of perspectives.

- **"Visual commonsense" that does not require domain knowledge**: The questions in BLINK are intentionally designed to be straightforward, requiring neither domain-specific knowledge nor expertise to answer. They are crafted in such a way that humans can solve them almost instantaneously, typically within a few seconds. This allows us to explore the fundamental gap in visual perception gap between humans and Multimodal LLMs, highlighting the paradox that problems easily solved by humans often pose significant challenges for machines.

- **Interleaved image-text formats**: BLINK features a heterogeneous question-answering format, wherein both questions and choices can be presented as text or images. This diversity compels Multimodal LLMs to genuinely understand the questions, pushing the boundaries of their interpretative capabilities.

- **Diverse image sources**: BLINK comprises a wide range of in-the-wild images sourced from various origins, covering everything from indoor and outdoor scenes to object-centric views and landscapes. This collection spans abstract diagrams, synthesized images, and authentic photographs, ensuring a comprehensive examination of visual perception

The design principles of BLINK are also illustrated in Figure 3.2. We will now describe each task in detail.

3.2.2 Dataset Collection Process

BLINK comprises 14 tasks, all of which have been repurposed into a multiple-choice question-answering format. These tasks utilize a diverse collection of images from various sources and we ensure that each image is unique.

Visual correspondence: This task aims to evaluate the ability of Multimodal LLMs to understand and identify the same scene point across various viewpoints, lighting conditions, or time. We exploit HPatches [Balntas et al., 2017] for this task. HPatches contains a number of image sequences, each of which are composed of images taken under different illuminations and/or viewpoints of a scene. For each question, we randomly sample two images and an interest point within them. Then we exploit the ground-truth homography to compute its correspondence. Finally, we randomly select three more interest points to serves as other choices.

Relative reflectance: This task aims to compare the reflectance (albedo) of two pixels. It allows us to evaluate Multimodal LLMs' understanding of material properties and their interaction with light, which is crucial for applications requiring high-fidelity visual interpretations. We curate our samples using human annotations from the Intrinsic Images in the Wild (IIW) dataset [Bell et al., 2014]. Each question is based on an image and two specified points. The goal is to identify which point is darker, or whether the two points have similar reflectance.

Relative depth: Humans are good at judging relative depth [Chen et al., 2016]. This task can thus serve as a proxy to validate if the geometric understanding capabilities of existing multimodal LLMs are close to human. We curate our samples using human annotations from the Depth in the Wild [Chen et al., 2016] dataset. Given an image and two specified points, the task is to determine which point is closer.

Figure 3.5: Qualitative results on BLINK. For each task, we show the choice of LLaVA-v1.6-34B [Liu et al., 2024b], Qwen-VL-Max [Bai et al., 2023], Gemini Pro [Team et al., 2023], GPT-4V [OpenAI, 2023b], and humans. Red choice indicates the ground truth. Notice that the markers are intentionally enlarged for visualization purposes, and we make some images inset images to save space. For IQ test, the third image is constructed by overlaying the first and second images.

Spatial relation: Understanding spatial relationships between objects in a scene is essential for interpreting complex visual environments. However, modern Multimodal LLMs often struggle with spatial concepts such as "left" and "right" [Yang et al., 2023b]. This task help us evaluate whether the models finally possess this vital skill. We curate our samples from the Visual Spatial Reasoning [Liu et al., 2023a] dataset. Each sample contains an image and a claim. The task is to determine if the claim is true or false. We reformat the claims into binary questions via GPT-3.5 [Brown et al., 2020].

Multi-view reasoning: This task is centered on evaluating the multi-view reasoning capabilities of Multimodal LLMs. The objective is to deduce the relative camera motion based on two images of an object captured from different viewpoints. Our data is sourced from the Wild6D dataset [Ze and Wang, 2022], which features videos of various objects recorded in diverse settings. We select two random frames from each video to calculate the relative camera motion. Recognizing that even humans might struggle to precisely articulate 3D motion details, we simplify the task by classifying motions into two broad categories: moving towards the left or moving towards the right. Despite the simplicity of these questions, as we will later demonstrate, they pose significant challenges for current models.

Jigsaw: This task assesses the ability of Multimodal LLMs to recognize and group patterns, as well as to align patches based on continuity in shape, color, and texture. We utilize images from the TARA dataset [Fu et al., 2022] and segment each of them into a 3x3 grid. We retain the three segments from the upper left corner as the reference image, and treat the central segment along with a randomly chosen segment as options. The objective is to identify the correct patch (*i.e.*, the central patch).

Art style: This task evaluates Multimodal LLMs capability to analyze and discern both local and global similarities in art styles among multiple images. Although there have been prior efforts to incorporate art-related questions into evaluation [Yue et al., 2024], such attempts primarily focused on questions requiring expert-level knowledge, including deducing an artist's name and understanding historical contexts, rather than on direct image comparison. For this task, we collect paintings and their stylistic information from WikiArt. Given one reference painting image and two other paintings as options, the model is tasked with identifying the one that most closely shares the art style of the reference painting.

Object localization: The ability to accurately detect and localize objects is critical for scene understanding. While previous benchmarks [Liu et al., 2024e] have explored this task, their focus was primarily on coarse localization. For instance, they might only ask the model if an object is located at the "top" or "right" side of an image. BLINK, in contrast, aims for a more fine-grained evaluation. We exploit images from LVIS [Gupta et al., 2019], randomly sampling one object per image along with its ground-truth bounding box. Then we add Gaussian noise to the ground-truth box to create a confounding box. The goal is to select the correct one.

Counting: This task evaluates Multimodal LLMs' abilities in detection, recognition, and compositional reasoning, particularly in complex scenes where objects may overlap, be occluded, or vary in size and appearance. We select our questions from the TallyQA dataset [Acharya et al., 2019], known for its challenging human-written counting questions. Each sample comprises an image, a question, and a numerical answer. In addition to the correct answer, we randomly select three numbers to serve as confounding options.

Forensic detection: Recent advances in generative AI have raised concerns about malicious uses and have prompted calls for the automatic detection of fake content. To evaluate whether Multimodal LLMs can fulfill such a role, we construct sets of real and synthesized images that describe similar scenes and ask the models to identify the real ones. Specifically, we first generate synthetic images using Stable Diffusion XL [Podell et al., 2023], employing COCO captions [Lin et al., 2014] as prompts. Then, we manually search online using these captions as descriptions and select high-quality photographs as the real images.

IQ test: This task evaluates the ability of Multimodal LLMs to engage in graphical reasoning, without requiring any domain-specific knowledge. We manually collect test samples, along with human explanations, from various public, license-friendly online sources. Given visual examples and a selection of images, the objective is to identify the image that either continues the pattern established by the examples or is spatially consistent with them.

Visual similarity: This task aims to verify whether Multimodal LLMs possess a nuanced understanding of visual features, patterns, and aesthetics at a level comparable to humans. We select our samples from the DreamSim dataset [Fu et al., 2023b]. Given a reference image alongside two alternative images, the objective is to identify the image that more closely resembles the reference image perceptually.

Semantic correspondence: This task focuses on identifying and matching semantically similar yet visually distinct elements across images, thereby evaluating the ability of Multimodal LLMs to understand the underlying semantics of object parts. Our samples are sourced from the SPair-71k dataset [Min et al., 2019], which features pairs of images with multiple corresponding semantic points. For each task, we randomly select one semantic point in an image as a reference, and provide the matching point alongside three random semantic points in the paired image as options. The objective is to accurately identify the correct matches.

Functional correspondence: The task aims to identify points that are functionally similar across objects. It challenges Multimodal LLMs to extend their understanding beyond mere semantics, enabling them to infer the diverse functions an object can perform in various contexts. Such capability is crucial for applications in robotics. We derive our samples from the FunKPoint dataset [Lai et al., 2021], which features paired images annotated for functional correspondences. Following a method analogous to semantic correspondence, we present an action alongside two object images. One image includes a reference point, while the other offers four potential points. The objective is to select the point that best matches the reference in terms of functional affordances.

Data quality control: To guarantee the quality of BLINK, we manually go through all collected data and filter out data that are ambiguous.

3.3 Experiments

In this section, we first describe the experimental setup and the baselines (§3.3.1). Then we present a comprehensive evaluation of 16 recent multimodal LLMs (§3.3.2). We demonstrate that while humans can answer the questions with high accuracy, BLINK is challenging for existing models. Finally, we provide detailed analyses on multiple experimental settings, including the effect of reducing images to captions, sensitivity to different visual prompts, and error analysis (§3.3.3).

3.3.1 Experimental Setup

Multimodal LLMs: We evaluate BLINK on 16 recent multimodal LLMs, including MiniGPT-4-v2 [Chen et al., 2023a], OpenFlamingo-v2 [Awadalla et al., 2023], InstructBLIP (7B and 13B) [Dai et al., 2023a], CogVLM [Wang et al., 2024a], LLaVA(v1, v1.5, v1.6, internLM, and xtuner versions, model size 7B, 13B, and 34B) [Liu et al., 2023b, 2024a,b; Dong et al., 2024; Contributors, 2023b], Yi-VL (6B and 34B), Qwen-VL-MAX [Bai et al., 2023], Gemini Pro [Team et al., 2023], Claude 3 Opus [Anthropic, 2024b] and GPT-4V(vision) [OpenAI, 2023b].

Evaluation setup: We follow standard setups as in the VLMEvalKit [Contributors, 2023a], where the temperature is set to 0 and retry is set to 10. However, we do not resize the images during any experiment. For the models that do not support multiple images as input, we concatenate the images as input. We extract the choice from the models' output with a set of pre-defined rules and GPT-3.5-turbo [Brown et al., 2020]. There are three types of visual prompts in BLINK: circles, boxes, and masks as shown in Figure 3.5. As for visual correspondence, functional correspondence, semantic correspondence, the red circles have radius 10px on images resized to 1024px height. For relative reflectance, we draw white circles to avoid color confusions. For object localization, the boxes are in red and green. For jigsaw, the masks are kept black.

3.3.2 Main Results

Overall performance: As shown in Table 3.1, the mean accuracy of 7B and 13B open-source Multimodal LLMs hover around 35–42%, which is similar to random guess (38.09%). The most proficient open-source model, LLaVA-v1.6-34B, achieves an accuracy of 45.05%. Even the most advanced models, GPT-4V and Gemini Pro and Claude 3 OPUS, achieve accuracies of only 51.26%, 45.72%, and 44.11% respectively. Their performance are merely 13.17%, 7.63% and 6.02% better than random guessing and lag behind human performance by 44.44%, 49.98% and 51.59%. Notably, for certain tasks such as jigsaw, semantic correspondence, multi-view reasoning, object localization, and relative reflectance, some multimodal LLMs even underperform compared to random guessing. Some qualitative results are shown in Figure 3.5.

In which tasks do multimodal LLMs show relative strengths and weaknesses? Figure 3.4 shows the accuracies of the best-performing models on BLINK: LLaVA-v1.6-34B [Liu et al., 2024b], Gemini Pro [Team et al., 2023], and GPT-4V [OpenAI, 2023b]. We observe that multimodal LLMs perform relatively better on spatial reasoning, art style, and counting tasks, in which they are much better than random guessing. The models also demonstrate some capability in relative depth and forensics detection. Overall, they are most effective on mid-level perception tasks. In terms of granularity, the models in general perform better on

	Validation (1,901)	Test (1,906)	Similarity (136)	Counting (120)	Depth (124)	Jigsaw (150)	Art (117)	Fun.Corr. (130)
Random Choice	38.09	38.09	50	25	50	50	50	25
Human	95.67	95.70	96.70	93.75	99.19	99.00	95.30	80.77
Open-source multimodal LLMs								
MiniGPT-4-v2 [Chen et al., 2023a]	34.23	34.57	52.94	10.83	49.19	26.00	47.86	18.46
OpenFlamingo-v2 [Awadalla et al., 2023]	39.18	38.32	55.15	21.67	54.03	46.00	52.14	36.15
InstructBLIP-7B [Dai et al., 2023a]	39.72	38.65	46.32	29.17	50.81	54.00	47.86	23.85
InstructBLIP-13B [Dai et al., 2023a]	42.24	39.58	46.32	30.83	50.00	54.00	50.43	22.31
LLaVA-internLM2-7B [Team, 2023]	37.71	36.06	52.94	52.50	52.42	34.67	30.77	23.08
Yi-VL-6B	38.72	41.24	46.32	46.67	56.45	50.00	53.85	23.85
Yi-VL-34B	41.68	42.78	50.00	58.33	53.23	54.00	46.15	**39.23**
LLaVA-v1.5-7B-xtuner [Contributors, 2023b]	39.36	40.81	46.32	53.33	50.81	54.00	47.86	23.85
LLaVA-v1.5-13B-xtuner [Contributors, 2023b]	42.00	41.31	46.32	45.00	54.03	53.33	47.86	26.15
CogVLM [Wang et al., 2024a]	41.54	39.38	46.32	38.33	50.81	52.67	49.57	23.85
LLaVA-v1.5-7B [Liu et al., 2024a]	37.13	38.01	46.32	43.33	51.61	11.33	47.86	21.54
LLaVA-v1.5-13B [Liu et al., 2024a]	42.66	40.55	46.32	50.00	47.58	54.00	47.86	20.77
LLaVA-v1.6-34B [Liu et al., 2024b]	46.80	45.05	46.32	**68.33**	64.52	56.67	47.01	30.77
API-based models								
Qwen-VL-Max [Bai et al., 2023]	40.28	41.94	51.47	55.83	58.87	3.33	37.61	28.46
Gemini Pro [Team et al., 2023]	45.16	45.72	55.88	65.00	50.00	54.00	49.57	32.31
Claude 3 OPUS [Anthropic, 2024b]	44.05	44.11	70.59	49.17	57.26	32.67	60.68	22.31
GPT-4V(ision) [OpenAI, 2023b]	51.14	51.26	**83.09**	60.83	58.87	62.67	78.63	31.54
GPT-4 Turbo [OpenAI, 2023b]	54.61	53.89	**83.09**	60.83	**66.94**	**66.00**	81.20	31.54
GPT-4o [OpenAI, 2023b]	**60.04**	**59.03**	65.44	51.67	64.52	58.00	**82.91**	**39.23**

	Sem.Corr. (140)	Spatial (143)	Local. (125)	Vis.Corr. (172)	Multi-view (133)	Reflect. (134)	Forensic (132)	IQ (150)
Random Choice	25	50	50	25	50	33.33	25	25
Human	96.07	98.25	98.00	99.42	92.48	95.14	100.00	80.00
Open-source multimodal LLMs								
MiniGPT-4-v2 [Chen et al., 2023a]	26.43	51.75	**56.00**	23.84	52.63	31.34	17.42	19.33
OpenFlamingo-v2 [Awadalla et al., 2023]	23.57	46.85	52.00	25.00	41.35	43.28	15.91	23.33
InstructBLIP-7B [Dai et al., 2023a]	25.00	55.24	44.80	22.67	**58.65**	29.85	29.55	23.33
InstructBLIP-13B [Dai et al., 2023a]	22.86	64.34	52.00	20.93	54.14	46.27	13.64	26.00
LLaVA-internLM2-7B [Team, 2023]	22.14	74.13	48.00	21.51	41.35	32.84	3.79	14.67
Yi-VL-6B	26.43	72.73	49.60	29.65	48.12	29.85	20.45	23.33
Yi-VL-34B	21.43	70.63	54.40	23.84	41.35	46.27	17.42	22.67
LLaVA-v1.5-7B-xtuner [Contributors, 2023b]	24.29	74.83	45.60	23.84	42.11	26.87	36.36	21.33
LLaVA-v1.5-13B-xtuner [Contributors, 2023b]	22.14	**77.62**	48.00	22.09	41.35	46.27	29.55	18.67
CogVLM [Wang et al., 2024a]	23.57	67.13	43.20	20.93	57.14	26.87	24.24	26.67
LLaVA-v1.5-7B [Liu et al., 2024a]	32.14	70.63	48.80	20.35	49.62	36.57	28.03	24.00
LLaVA-v1.5-13B [Liu et al., 2024a]	23.57	67.83	47.20	20.35	41.35	45.52	27.27	28.00
LLaVA-v1.6-34B [Liu et al., 2024b]	27.86	76.22	41.60	27.33	46.62	29.85	41.67	26.00
API-based models								
Qwen-VL-Max [Bai et al., 2023]	29.29	**77.62**	49.60	22.67	53.38	**49.25**	47.73	22.00
Gemini Pro [Team et al., 2023]	22.14	67.13	46.40	37.21	41.35	46.27	45.45	27.33
Claude 3 OPUS [Anthropic, 2024b]	20.71	57.34	46.40	31.40	57.89	27.61	62.12	21.33
GPT-4V(ision) [OpenAI, 2023b]	30.00	72.03	50.40	37.21	58.65	38.81	30.30	24.67
GPT-4 Turbo [OpenAI, 2023b]	32.86	67.13	48.80	42.44	57.14	34.33	51.52	**30.67**
GPT-4o [OpenAI, 2023b]	**45.71**	76.92	**56.00**	**71.51**	**60.15**	38.81	**85.61**	30.00

Table 3.1: Results of different models on the BLINK test set. The first row shows task names and number of test data. The best performance in each task is in-bold. For the sake of completion, we also show the average score on the validation set.

image-level tasks and struggle on pixel-level and crop-level tasks.

GPT-4V behaves differently: Figure 3.4 and Table 3.1 show an interesting phenomenon: GPT-4V's performance pattern is different from other models. Compared with its counterparts, GPT-4V is much better in visual similarity, art style, jigsaw, and multi-view reasoning. Specifically, its performance on visual similarity is 29% better than Gemini Pro, demonstrating that GPT-4V possesses a nuanced understanding

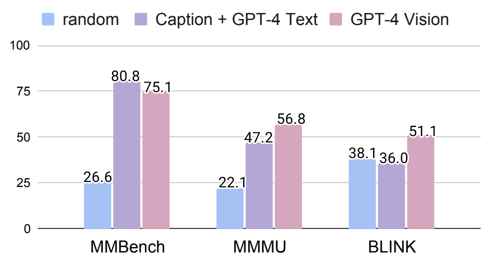


Figure 3.6: Performance of using image caption + text-only GPT-4 *vs.* GPT-4 Vision on MMBench Liu et al. [2024e], MMMU Yue et al. [2024], and BLINK (§3.3.3).

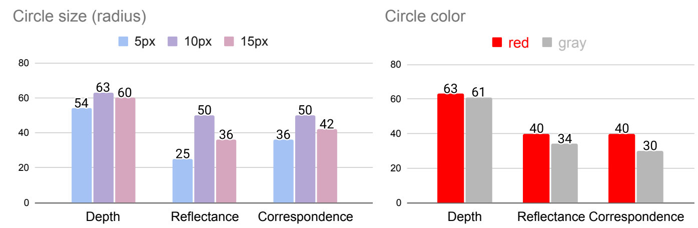


Figure 3.7: Accuracy of GPT-4V with different visual prompts (*e.g.*, different circle sizes, colors) on relative depth, relative reflectance, and visual correspondence tasks. More discussions in §3.3.3.

of visual patterns and aesthetics that is similar to humans. In contrast, Gemini Pro and LLaVA have similar performance patterns.

Human performance: Human evaluators achieve over 95% accuracy across most tasks, with an average accuracy of 95.70%.[2] This performance disparity between humans and multimodal LLMs highlights the significant visual perception gap that exists between current machine learning models and humans in perceiving, processing, and understanding complex visual and textual context.

3.3.3 Analysis

Is dense captioning all you need for a multimodal LLM benchmark? To answer the question, we reduce multimodal benchmarks to a text-only problem. Specifically, we convert images into task-agnostic dense image captions with GPT-4V. The dense caption describes detailed information about the image and the visual prompts (*e.g.*, where each circle is), using language. For each multimodal question, we prompt the text-only GPT-4-0125-preview model with image captions and the textual question and evaluate if the "blind" GPT-4 can answer the question. We call this `Caption + LLM`. This experiment is predicated on the hypothesis that captioning involves predominantly recognition-centric perception. If using captions along with text-only LLMs yields performance comparable to or surpassing that achieved through the integration of images with multimodal LLMs, then the perception demands of that benchmark are primarily confined to recognition

[2]Note that the human score for IQ test is annotated by authors. It may not reflect typical human performance, which is also expected to vary.

Task	Vis.Corr.	Depth	Multi-view	Sem.Corr.
Random	25.00	50.00	50.00	25.00
Human	99.56	99.59	92.10	94.60
Gemini Pro	42.44	40.32	44.36	26.62
GPT-4V	33.72	59.68	55.64	28.78
Specialist	DIFT Tang et al. [2023] 96.51	DepthAnything Yang et al. [2024b] 97.58	LoFTR Sun et al. [2021] 90.22	DIFT Tang et al. [2023] 71.22

Table 3.2: Comparison between multimodal LLMs, specialists, and human performance on the BLINK dev set. The specialists perform much better than multimodal LLMs.

only.

We experiment with BLINK, MMBench [Liu et al., 2024e] and MMMU [Yue et al., 2024], as illustrated in Figure 3.6. Surprisingly, we find that the `Caption + LLM` setting achieves better results on MMBench than GPT-4V (with 5.7% increase in accuracy). On MMMU, `Caption + LLM` achieves 47.2% accuracy, which is 9.6% lower than GPT-4V performance, but is still much better than random guessing. On BLINK, `Caption + LLM` fails, achieving random guessing performance. These results indicate that dense captions cover the visual information needed for MMBench. For MMMU, image captions carry a large portion of visual information needed to answer the domain-knowledge-specific questions. Meanwhile, the performance decrease observed in BLINK suggests the necessity or advanced perceptual abilities beyond what is currently attainable with general captions. This variance highlights the limitations of existing multimodal LLM benchmarks in addressing the full spectrum of visual perception.

Effect of visual prompting on BLINK: Several BLINK tasks involve visual prompting. Prior work [Shtedritski et al., 2023] shows that factors like shape, size, and color may affect task performance, and circles give the best overall performance. Following [Shtedritski et al., 2023], we adopt circles in BLINK and analyze the effect of circle sizes and colors on multiple tasks in Figure 3.7. We experiment with relative depth, relative reflectance, and visual correspondence, with 100 validation set samples per task. The images are all reshaped to 1024px height. We experiment with circles with 5px, 10px, and 15px radius, and with red or gray color. We find that red is better than gray for all tasks. Also, the optimal circle size is task-dependent. On average 10px circles work the best, and we use it for all evaluations. The experiments suggest that visual prompting can have a big impact on multimodal LLM performance, and improving visual prompts or improving model robustness to prompt variation is a promising direction for future research [Yang et al., 2023a].

Can specialist models solve BLINK tasks? Specialists can serve as a proxy upper bound of how good multimodal LLMs could be. We download the trained checkpoints for six specialist models and evaluate them on BLINK. As shown in Table 3.2, the specialists perform much better than GPT-4V and Gemini Pro, outperforming the best multimodal LLM by 18% to 57% on these tasks. Specifically, DepthAnything [Yang et al., 2024b] and DIFT [Tang et al., 2023] achieve human-level performance on depth estimation and visual correspondence, whereas multimodal LLMs fail miserably. This sheds light on the possibility that multimodal LLMs may progress on these tasks given the correct data and training strategy. For instance, one possible way is to distill existing specialist models into multimodal LLMs [Hu et al., 2024b].

Error analysis of GPT-4V: We randomly sampled 140 error instances made by GPT-4V on BLINK, 10 per task, and meticulously examined them. The most common types of errors are:

- **Hallucinate fine-grained patterns and attributes** (24.2%): the model hallucinates the nuanced details of objects. This error is most common for relative reflectance, forensics detection, and jigsaw tasks.
- **Hallucinate visual prompt locations** (20.0%): the circle location described by the model is wrong. This is common for visual correspondence and relative depth tasks.

Other errors include Failures on capturing overall setting or style (8.6%), and Failures on grounding an object (5.7%).

3.4 Summary

We introduced BLINK, a new multimodal LLM benchmark that evaluates core visual perception abilities not found in existing evaluations. While these tasks seem trivial for humans to solve "within a blink," we find they pose significant challenges for current multimodal LLMs. Even the powerful GPT-4V and Gemini models only achieve around 50% accuracy on BLINK, far below the 95.7% human performance. We conduct extensive analysis, measuring the effect of converting images to dense captions, visual prompting, self-consistency, analyzing the capabilities of specialist models, and conducting error analysis. We highlight that specialist computer vision models are performing much better than GPT-4V and Gemini on BLINK, shedding light on the possibility that multimodal LLMs may have big progress on these tasks. Ultimately, BLINK provides a simple yet effective testbed for multimodal LLMs to catch up with human-level visual perception.

Chapter 4

Visual Sketchpad: Sketching as a Visual Chain of Thought for Multimodal Language Models

4.1 Introduction[1]

Sketching is a fundamental human activity, serving as a versatile tool for communication [Goel, 1995], ideation [Tversky et al., 2003], and problem-solving [Tversky and Suwa, 2009]. Unlike written language, sketches have the advantage of conveying visuo-spatial ideas directly, for example by using spatial relations on paper to convey spatial relations or other more abstract relationships in the world. This may explain their ubiquity; maps [Taylor and Tversky, 1992b] and architectural plans [Goldschmidt, 1991] have been found incised in stone, etched on leather, impressed in clay, and drawn on paper in diverse cultures scattered across the world [Taylor and Tversky, 1992a]. Sketches are so fundamental that we use them to teach school children how to solve geometry problems by drawing support lines, to aid engineers conveying prototypes, to support architects creating blueprints, and to allow scientists like us to convey scientific contributions (see Figure 4.1).

[1] This work involved several authors, and the primary authors are Weijia Shi and me. I proposed the idea, built the agent framework, and conducted all the experiments on visual reasoning. Weijia ran the experiments of math tasks. We wrote the paper together.

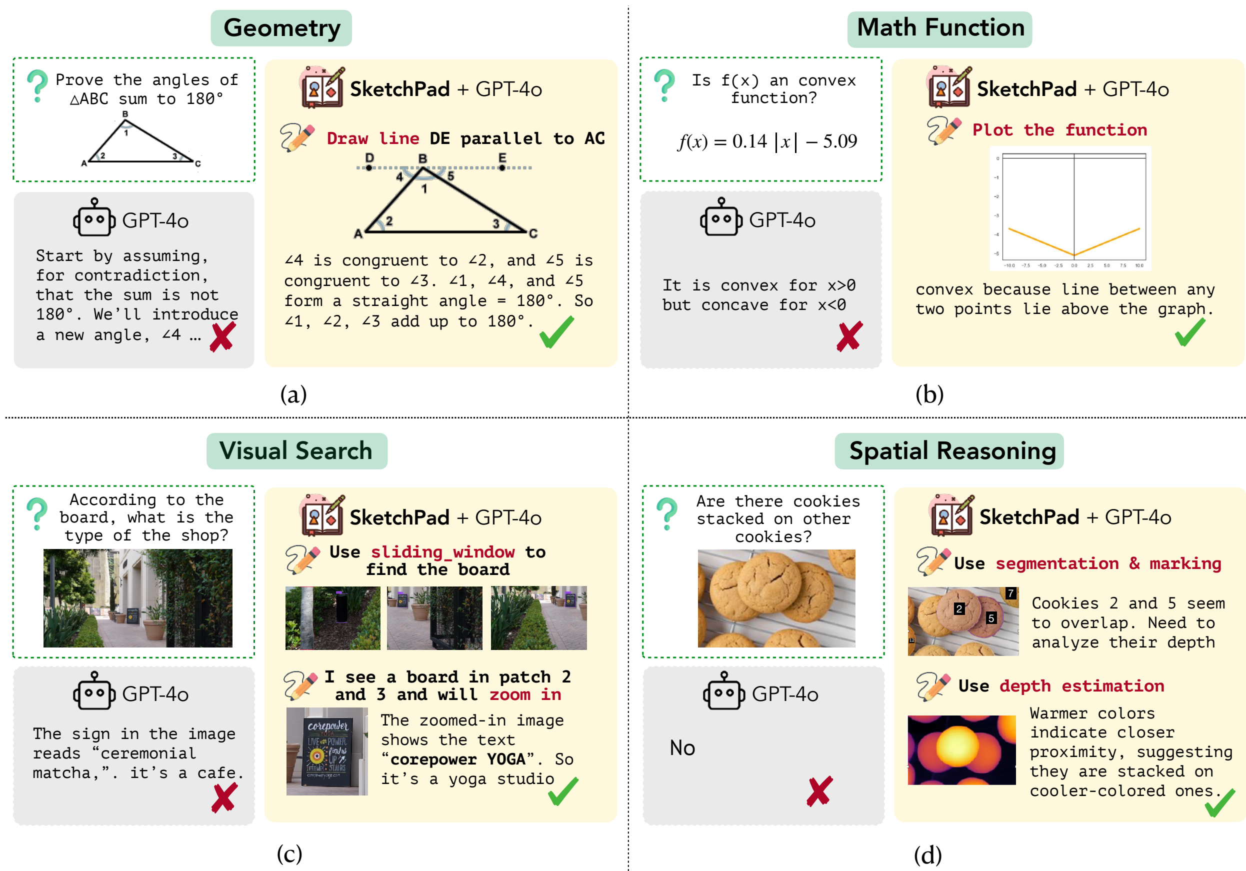


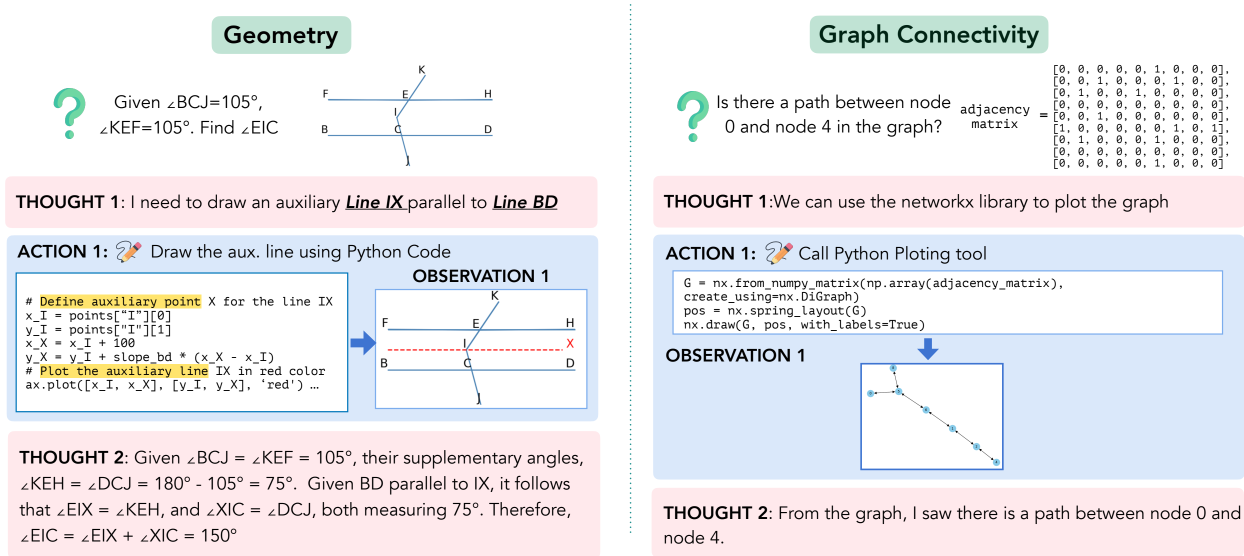
Figure 4.1: Sketchpad equips GPT-4 with the ability to generate intermediate sketches to reason over tasks. Given a visual input and query, such as proving the angles of a triangle equal 180°, Sketchpad enables the model to draw auxiliary lines which help solve the geometry problem. The examples are from [Fu et al.; Wu and Xie, 2024; Tong et al., 2024]. For all these examples, without Sketchpad, GPT-4o fails to get the correct answer, while Sketchpad + GPT-4o achieves the correct solution.

As multimodal language models [OpenAI, 2023b; Team et al., 2023; Liu et al., 2023b, 2024b; Alayrac et al., 2022; Bai et al., 2023; Wang et al., 2024a; Dai et al., 2023a; Lu et al., 2024a; Team, 2024; Chen et al., 2024, 2023c] have begun to mature, we now expect them to solve tasks like the ones mentioned above, i.e., ones where people draw intermediate sketches to simplify reasoning. Popular benchmarks now include questions about geometry (e.g., Geometry3K [Lu et al., 2021]) and complex math problems (e.g., IsoBench [Fu et al.]). In these benchmarks, models are given images of diagrams and asked questions requiring symbolic grounding and spatial understanding, where intermediate sketches like auxiliary lines can enhance reasoning. Even benchmarks in computer vision now have a similar flavor. Specialist vision models can be viewed as sketching on natural images. For example, object detection is plotting bounding boxes around objects; depth estimation is drawing colormaps according to depth. The BLINK benchmark would benefit significantly from such intermediate visual sketches. Similarly, the V^*Bench benchmark [Wu and Xie, 2024] could focus reasoning on image crops to find answers. Unfortunately, current LMs lack a scaffold for using sketch-based reasoning when solving tasks.

We introduce Visual Sketchpad, a framework that provides multimodal LMs with the tools necessary to generate intermediate sketches to reason over tasks. Inspired by textual chain-of-thought reasoning in LMs [Wei et al., 2022; Zhang et al., 2024b], Sketchpad prompts the underlying visual LM to produce visual artifacts as part of a chain of mixed textual, programmatic, and visual reasoning. For example, to prove that the angles of triangles sum up to 180 degrees in Figure 4.1 (a), Sketchpad enables agents to modify the diagram by introducing a new auxiliary line. This new line, along with new annotated angles, provides the critical information to solve the geometry task. Similarly, Sketchpad improves models' spatial reasoning for computer vision. To determine if there are cookies stacked on top of other cookies in the image (Figure 4.1b), the model first produces an intermediate depth estimate. By analyzing the depth estimate, which reveals cookies overlapping at different depths, the model is able to correctly answer that the cookies are indeed stacked.

We demonstrate the effectiveness of visual Sketchpad across a wide range of mathematics and computer vision tasks. For math, we tackle problems including (1) geometry [Lu et al., 2021], (2) mathematical functions, (3) graph algorithms, and (4) strategy games [Fu et al.]. For geometry questions, Sketchpad enables models to generate Matplotlib code with auxiliary lines and variables, given the diagram input and questions (Figure 4.1a). Notably, even when the input is solely language-based, such as mathematical functions, Sketchpad enables models to plot the functions and reason about their properties, using only the mathematical function expression as input (Figure 4.1b). These results highlight the ability of Sketchpad to aid reasoning, even in tasks with purely language-based inputs. **Across all four categories of mathematical tasks, Sketchpad consistently improves the baseline GPT-4o performance, yielding an average gain in accuracy of 11.2%.**

For computer vision, we tackle diverse tasks including (1) depth, (2) spatial reasoning, (3) jigsaw, (4) visual correspondence, (5) semantic correspondence, as well as questions from (6) the MMVP and (7) the V^*Bench benchmarks [Fu et al., 2024; Tong et al., 2024; Wu and Xie, 2024]. For this domain, Sketchpad enables models to generate segmentation masks, crop images, draw bounding boxes, zoom into image regions,



(a) (b)

Figure 4.2: Overview of Sketchpad. Given a multimodal query, the Sketchpad agent generates a sketching plan to address the query (*Thought*), and then synthesizes a program to create visual sketches (*Action*). By analyzing the resulting sketches (*Observation*), which serve as a visual representation of the reasoning process, the model generates a final response to the query.

overlay images, etc. Similar to math, **Sketchpad shows consistent improvements across all seven types of computer vision tasks**. For example, GPT-4o, augmented with Sketchpad, sees 14.3% absolute improvement in accuracy on *V**Bench, 12.1% and 9.7% improvements on BLINK's depth and semantic correspondence tasks, respectively, setting a new state of the art across all tasks. Finally, we analyze the effectiveness of Sketchpad by comparing the plans generated by our model with human-created plans, showing that they are well-aligned and exhibit similar reasoning patterns. We hope Sketchpad opens up new research opportunities toward more capable and interpretable multimodal intelligence.

4.2 Visual Sketchpad

We introduce visual Sketchpad, a general framework that enables multimodal LMs to draw sketches as intermediate reasoning steps and to use these sketches to facilitate further reasoning. Figure 4.2 shows examples of how Sketchpad works. Given a multimodal query, Sketchpad agent generates a sketching plan to address the query (*Thought*), and then synthesizes a program to create visual sketches (*Action*). By analyzing the resulting sketches (*Observation*), which serve as a visual representation of the reasoning process, the model generates a final response to the query.

Our framework requires no finetuning or training. Multimodal LMs, out of the box, can be prompted to sketch using our framework. Our implementation is based on the AutoGen [Wu et al., 2023] framework. We give the overview of our Sketchpad framework in §4.2.1, and describe how it integrates sketching into the reasoning process in §4.2.2.

4.2.1 Overview of Sketchpad

The Sketchpad agent solves tasks by engaging in an iterative interaction process with an environment. Given a multimodal query q that includes both visual and textual components, the model generates a series of thoughts, actions, and observations to gather the information needed to answer the query. At each time step t, the model performs three key steps:

Thought: The model analyzes the current context c_t, which includes the query, previous thoughts, actions, and observations, to generate a thought plan p_t for the next action. For example, given the query q – "*find the* $\angle EIC$" in Figure 4.2a, the model's thought plan p_1 is to draw an auxiliary line IX parallel to BD serving as a ***visual sketch*** to help solve the problem.

Action: Based on the thought plan, the model executes an action a_t, which can manipulate both visual and textual content. In the geometry example, to realize the proposed thought of drawing the auxiliary line, the model generates Python code to modify the original geometry diagram. The generated code is then compiled and executed.

Observation: Based on the action a_t, Sketchpad's environment returns a new observation o_{t+1}, such as a new diagram with the auxiliary line drawn in the geometry example. The multimodal context is then updated to $c_{t+1} = (c_t, p_t, a_t, o_{t+1})$.

The multi-turn interaction process continues until time step T, when the model determines that it has gathered enough information from the context c_T to answer the query. At this point, it generates a special **Terminate** action and provides the answer.

Different from prior work [Yao et al., 2023], where LMs primarily generate and manipulate text-based observations and actions, Sketchpad enables the model to work with **multimodal observations o_t and actions a_t, manipulating both visual and textual content.** This allows the model to plan and reason with the visual sketches it has drawn, enhancing its problem-solving capabilities.

4.2.2 Sketching via Code Generation

The core component of Sketchpad is sketching, which enables the LM to generate visual sketches by synthesizing programs that call different specialist vision models or Python plotting packages.

Program Generation. Similar to recent works like ViperGPT and VPD [Gupta and Kembhavi, 2023; Surís et al., 2023; Hu et al., 2024b], Sketchpad enables LMs to sketch through code generation. The LM is provided, through a prompt, with a detailed description of the available tools that can generate multimodal content (an example prompt and description can be found in §A.1). The prompt includes Python function signatures and docstrings [Hsieh et al., 2023] for these modules, but does not contain their full implementation. The LM generates Python code in a code block, using the provided tools, which, when executed, generates new image and text outputs. A special *display* function allows the LM to **visualize** the sketch image in the next observation o_{t+1}.

Modules for sketching. Sketchpad uses various tools to facilitate the sketching process, depending on the task at hand. For mathematical tasks, Sketchpad uses common Python packages like `matplotlib` and `networkx` for plotting (see §4.3). For vision tasks, the LM leverages **specialist vision models** during the sketching process. These models include detection tools that draw bounding boxes on the image, as well as segmentation and marking tools (inspired by SoM [Yang et al., 2023a]) that draw colorful masks on the image and use numbers to label segments. We find these specialists possess useful perception skills for visual reasoning tasks, and Sketchpad is an effective way to combine them into a multimodal LM (see §4.4.1).

4.3 Sketching to Solve Math Problems

In this section, we experiment with Sketchpad on four complex mathematical tasks: (1) geometry, (2) mathematical functions, (3) graph algorithms, and (4) game strategies. We demonstrate that incorporating sketching capabilities into LMs significantly improves their performance on these mathematical problems, setting new state-of-the-art results (§4.3.2).

4.3.1 Math tasks

Details of our evaluation tasks and the tools employed for visual reasoning are as follows:

Geometry Problems. Drawing auxiliary lines in geometry diagrams is often helpful for problem-solving. For example, in Figure 4.2 (a), when asked to find $\angle EIC$, the LM plans to draw an auxiliary line IX parallel to BD, allowing it to use the properties of parallel lines to determine $\angle EIC$. To evaluate the effectiveness of Sketchpad, we use the problems from the Geometry3K dataset [Lu et al., 2021].

To realize the line drawing process, Sketchpad takes a geometry diagram and its corresponding `matplotlib` code as input. The model then proposes and modifies the code to generate auxiliary lines, and executes the modified code to visualize the updated diagram with the added lines.

Mathematical functions. Understanding mathematical functions is crucial for various applications in science, engineering, and economics. We focus on two tasks related to mathematical functions from the IsoBench datasets [Fu et al.]:

- ***Classifying parity*** aims to determine whether a function is even, odd, or neither. Even functions satisfy $f(-x) = f(x)$ for all x, while odd functions satisfy $f(-x) = -f(x)$.
- ***Identifying convexity/concavity*** aims to determine whether a function is convex or concave.

Existing LMs can only analyze functions and attempt to prove their properties analytically.[2] However, Sketchpad enables them to visually sketch functions to solve problems more efficiently. For instance, to determine the convexity of the function in Figure 4.1b, Sketchpad allows the model to plot the function using `matplotlib`, and visually inspect its overall shape.

Graph algorithms. Many real-world problems, such as those related to computer networks and transportation systems, can be formulated as graph problems. We evaluate Sketchpad on three graph problems from IsoBench [Fu et al.]:

- ***Graph connectivity*** determines whether there exists a path between two vertices in a graph.

[2] For humans, the analytical approach is the correct way to tackle these math tasks. However, we observe that LMs are not good at analytical reasoning in math. They make errors when deducing $f(-x)$ and derivatives.

Model	Geometry	Graph			Math		Game
	Geometry	Maxflow	Isomorphism	Connectivity	Convexity	Parity	Winner ID
Prior LLMs without visual inputs							
Gemini-Pro	\	15.6	47.7	50.0	87.9	48.2	8.1
Claude 3 OPUS	\	56.3	50.0	82.0	93.0	77.6	74.4
Mixtral 8x7B [Jiang et al., 2023]	\	8.6	50.0	62.5	69.1	41.7	7.4
LLaMA-2-70B [Touvron et al., 2023]	\	18.0	50.0	50.0	74.2	33.3	12.4
Latest multimodal LLMs + Visual Sketchpad							
GPT-4 Turbo	37.5	32.8	62.5	66.0	57.0	80.5	50.4
+ Sketchpad	45.8	96.8	97.6	97.6	77.3	71.5	64.2
	+8.3	+64.0	+35.1	+31.6	+20.3	-9.0	+13.8
GPT-4o	62.5	25.0	50.8	96.1	87.2	84.4	61.1
+ Sketchpad	**66.7**	**66.3**	**65.3**	**98.4**	**94.9**	**94.7**	**64.6**
	+4.2	+41.3	+14.5	+2.3	+7.7	+10.3	+3.5

Table 4.1: Accuracy scores on geometry problems, graph algorithms, mathematical functions, and game. **Sketchpad yields large performance gains on most tasks and outperform all baselines.**

- ***Maximum flow*** aims to find the maximum amount of flow that can be sent through a network from a source vertex to a sink vertex, subject to capacity constraints on the edges.

- ***Graph isomorphism*** tests whether two graphs are structurally equivalent.

Given an adjacency matrix of a graph like in Figure 4.2(b), Sketchpad can draw the actual graph structure, using using Python's `networkx` library, enabling direct visual reasoning about graph properties and relationships.

Game strategies. Chess games can be represented in various formats, including visual board states and textual move notations. Given only the textual move notations, Sketchpad can draw the visual representations of the chess board to analyze positions and formulate strategies. We evaluate the performance of Sketchpad on the winnder identification task from the IsoBench datasets [Fu et al.] that aims to find the outcome of a chess game (win for White, win for Black, or draw) based on the final board state. To create the graphical board, Sketchpad uses Python's `chess` library to draw the board from the Forsyth-Edwards Notation (FEN) of chess.

4.3.2 Results

We evaluate the performance of Sketchpad on multimodal LMs with API access, including gpt-4-turbo-2024-04-29 and gpt-4o-2024-05-13. We compare these results to baselines without the Visual Sketchpad and other notable closed-source models, such as Claude 3 and Gemini-Pro, as well as open-source models like Mistral [Jiang et al., 2023] and LLaMA-2 70B [Touvron et al., 2023].

Main results. As shown in Table 4.1, Sketchpad consistently improves base model performance across all tasks, with an average improvement of 11.2% for `GPT-4o` and 23.4% for `GPT-4 Turbo`. In particular, we observe large gains on graph algorithms such as maximum flow and connectivity. For instance, `GPT-4o` with Sketchpad achieves an accuracy of 66.3% on the maximum flow problem, improving over the base model by 41.3%. Similarly, Sketchpad substantially improves the performance on mathematical functions, with `GPT-4 Turbo` achieving over 90% accuracy and `GPT-4o` over 88% accuracy on convexity and parity classification tasks. Furthermore, we observe gains (3% ~ 10%) on game strategies, demonstrating that drawn game boards drawn can improve reasoning about game strategies. Overall, these results highlight the effectiveness of Sketchpad in enhancing the reasoning capabilities of multimodal language models across diverse domains.

4.4 Sketching to Solve Computer Vision Tasks

In this section, we experiment with Sketchpad on complex visual reasoning tasks. Experiments on BLINK (§3.3.3) finds that many core visual perception abilities are still missing from existing multimodal LMs—even though many computer vision specialist models possess such abilities. Also, SoM [Yang et al., 2023a] shows that drawing segmentation masks on images unleashes the strong visual grounding ability of GPT-4V. We generalize these ideas with Sketchpad, allowing LMs to use **specialist vision models** to sketch. Details of these modules are in §4.4.1. Sketchpad enhances multimodal LMs' visual reasoning abilities and establishes new SOTAs on all 7 tasks (§4.4.2).

Tasks. We experiment with a wide range of complex visual reasoning tasks: (1) V^***Bench** [Wu and Xie, 2024]. This benchmark contains questions about small items in an image. (2) **MMVP** benchmark from *Eyes Wide Shut* [Tong et al., 2024]. This benchmark contains visual questions specially designed to reveal the visual shortcomings of CLIP-based multimodal LMs. (3) **BLINK** [Fu et al., 2024]. This benchmark contains visual perception tasks that are easy for humans, but post significant challenge for multimodal LMs. Specifically, we experiment with relative depth, spatial reasoning, jigsaw puzzle, visual correspondence, and semantic correspondence tasks. More details of each task are in §A.2.

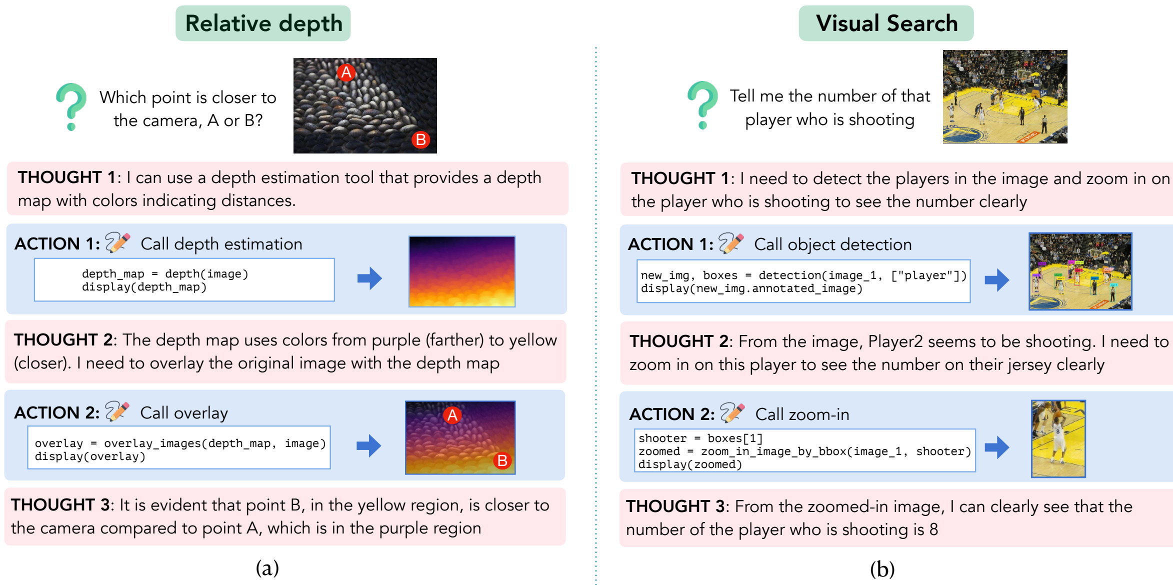


Figure 4.3: Examples of Sketchpad applied to vision tasks. The figure shows actual outputs generated by Sketchpad. By contrast, the baseline GPT-4o model cannot answer these questions correctly. Note that for demonstration purposes, the "A" and "B" marks in (a) are different from the actual images in the experiments.

4.4.1 Vision Specialists as Sketching Tools in Sketchpad

LMs can use the following modules to sketch and manipulate images. We wrap these modules into Python functions that the LMs can call. Refer to §A.1 for the function definitions.

Detection. This module takes an image and a simple text query (e.g., "cat") as input. We run the Grounding-DINO [Liu et al., 2024d] open-vocabulary objection detection model and plot the detected bounding boxes (together with a number label) on the image. It also returns the bounding box coordinates.

Segmentation. This module takes an image as input and returns an image with colorful segmentation masks on it. Each mask also has a number label. We follow the implementation of SoM [Yang et al., 2023a]. The underlying segmentation models are SegmentAnything [Kirillov et al., 2023] and Semantic-SAM [Li et al., 2024c].

Depth estimation. This module takes an image as input and returns a depth map. The underlying model is DepthAnything [Yang et al., 2024b].

Visual search via sliding window. This module mimics how humans search for small items on an image. It takes a text query as input and runs a sliding window over the image. The window size is 1/3 of the image size, and the step size is 2/9 of the image size (so an image will have $4 \times 4 = 16$ windows). It returns a sequence of image patches in which the query is detected.

Other image manipulation modules. Other modules include (1) **zoom-in and crop**, which takes an image and a bounding box as input and returns the image patch inside the box; (2) **Overlay images**, which takes two images and alpha values as input, and returns the overlayed image.

4.4.2 Results

We experiment with the same multimodal LMs as in §4.3 on complex visual reasoning tasks. We compare the performance with and without Sketchpad, as well as other notable multimodal LMs, including Gemini [Team et al., 2023], Claude 3 [Anthropic, 2024b], and the open-source LLaVA 1.5 [Liu et al., 2024a], LLaVA-NeXT [Liu et al., 2024b].

Model	V^*Bench	MMVP	Depth	Spatial	Jigsaw	Vis. Corr.	Sem. Corr.
Prior multimodal LLMs							
LLaVA-1.5-7B [Liu et al., 2024a]	48.7	-	52.4	61.5	11.3	25.6	23.0
LLaVA-1.5-13B [Liu et al., 2024a]	-	24.7	53.2	67.8	58.0	29.1	32.4
LLaVA-NeXT-34B [Liu et al., 2024b]	-	-	67.7	74.8	54.7	30.8	23.7
Claude 3 OPUS [Anthropic, 2024b]	-	-	47.6	58.0	32.7	36.6	25.2
Gemini-Pro [Team et al., 2023]	48.2	40.7	40.3	74.8	57.3	42.4	26.6
GPT-4V-preview [OpenAI, 2023b]	55.0	38.7	59.7	72.7	70.0	33.7	28.8
Latest multimodal LLMs + Visual Sketchpad							
GPT-4 Turbo	52.5	71.0	66.1	68.5	64.7	48.8	30.9
+ Sketchpad	71.0	73.3	68.5	80.4	68.5	52.3	42.4
	+18.5	+2.3	+2.4	+11.9	+3.8	+3.5	+11.5
GPT-4o	66.0	85.3	71.8	72.0	64.0	73.3	48.6
+ Sketchpad	**80.3**	**86.3**	**83.9**	**81.1**	**70.7**	**80.8**	**58.3**
	+14.3	+1.0	+12.1	+9.1	+6.7	+7.5	+9.7

Table 4.2: Accuracy on complex visual reasoning tasks. **Sketchpad enhances both GPT-4 Turbo and GPT-4o performance, establishing new SOTA performance levels on all the tasks.**

Main results. Table 4.2 shows the performance of our Sketchpad and baselines. Sketchpad consistently improves base model performance across all tasks. GPT-4o with Sketchpad sets the new state-of-the-art results on all tasks. Sketchpad is particularly effective on V^*Bench, yielding 18.5% accuracy improvement for GPT-4 Turbo and 14.3% improvement for GPT-4o, surpassing the previous state of the art SEAL [Wu and Xie, 2024] which used a visual search model specifically trained for this task. On BLINK tasks, Sketchpad on average yields 6.6% absolute accuracy gain for GPT-4 Turbo and 9.0% gain for GPT-4o. Interestingly, despite the fact that all modules in Sketchpad work on a single image, the LMs also get substantial improvement on multi-image tasks, including jigsaw puzzles, visual correspondence, and semantic correspondence. Finally, GPT-4o, the LM with stronger multimodal ability than GPT-4 Turbo, benefits more from Sketchpad. For example, on the relative depth task, GPT-4o gets 12.1% accuracy improvement, while GPT-4 Turbo only gets 2.4%, showing that GPT-4o is better at understanding the depth map Sketchpad generated. Overall, our experiments show that Sketchpad is an effective way to improve multimodal LMs' performance on visual reasoning tasks.

How many times is each vision specialist used? We count the number of times each vision specialist is used in each task, as shown in Figure 4.4. Here we choose the four tasks that achieve the largest improvement: V^*Bench, relative depth, spatial reasoning, and semantic correspondence. We observe that (1) **the use of vision specialist is task-dependent, and the two LMs analyzed utilize similar tools.** For example, for V^*, which needs to locate small objects, the LMs mainly use detection, sliding window search, and zoom-in, similar to how people would search. For the relative depth task, both models rely on depth estimation. For spatial reasoning, the LMs use detection and segmentation to facilitate visual reasoning. (2) **GPT-4o likes to use more tools.** GPT-4o uses the vision specialists more often than GPT-4 Turbo. Also, the two LMs behave differently for the semantic correspondence tasks. GPT-4o uses the segmentation module for 40% of the task instances, while GPT-4 Turbo uses the detection module for less than 20% of times, and rarely uses the segmentation module. This difference may explain the performance gap between the two LMs (58.3% v.s. 42.4%) on this task.

Comparison with visual prompting and tool-use frameworks. In Table 4.3, we compare Sketchpad with the visual prompting framework **SoM** [Yang et al., 2023a] and the LLM tool-use framework **Visprog** [Gupta

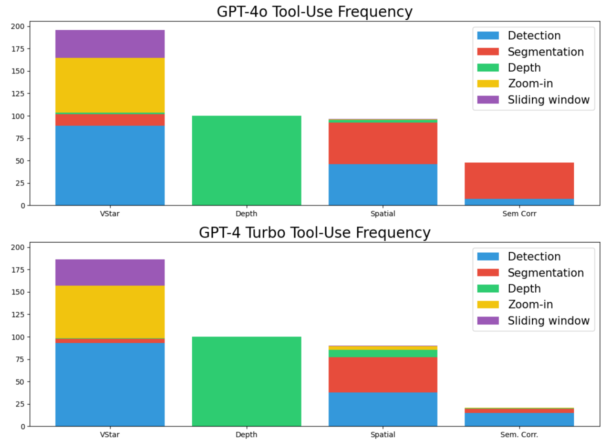


Figure 4.4: Percentage of times GPT-4o and GPT-4 Turbo use a visual module in Sketchpad when solving V^*Bench, relative depth, spatial reasoning, and semantic correspondence tasks.

Model	V^*	MMVP	Depth	Spatial
GPT-4 Turbo	52.5	71.0	66.1	68.5
SoM	42.0	60.7	58.9	78.3
SoM + orig.	51.3	**74.3**	66.9	79.7
Visprog	33.2	16.3	67.8	53.8
Sketchpad	**71.0**	73.3	**68.5**	**80.4**
GPT-4o	66.0	85.3	71.8	72.0
SoM	49.0	70.7	62.9	**83.2**
SoM + orig.	68.1	84.0	75.0	82.5
Visprog	32.4	17.3	46.8	37.8
Sketchpad	**80.3**	**86.3**	**83.9**	81.1

Table 4.3: Comparison with other augmentation frameworks for multimodal LMs on single-image tasks. For fair comparison, we modify the original Visprog Gupta and Kembhavi [2023] framework by replacing the LM and VQA components with the corresponding GPT-4 model.

and Kembhavi, 2023].For a fair comparison, we make the following adaptations: (1) we find that prompting LMs with SoM images can hurt performance, likely because the visual prompts confuse the model. To make a stronger baseline, we prompt the LM with both the original image and the SoM image (full prompt in §A.1), which we refer as "SoM + orig." (2) We replace the LM and VQA modules in Visprog with the corresponding GPT-4 model. (3) Since baseline methods are developed on single-image tasks, we compare Sketchpad on such tasks. From Table 4.3, we can see that **Sketchpad is the only framework that yields consistent improvement on all tasks.** SoM can boost spatial reasoning ability, as the authors reported. However, it can hurt the performance on other tasks, even in the "SoM + orig." setting. Visprog performs worse than the base LM on all the tasks. As prior work [Khandelwal et al., 2023; Hu et al., 2024b] suggests, one possible reason is that the vision modules themselves have errors, and the error propagates when the modules are composed by a program.

4.5 Analysis and Discussion

Why does Sketchpad work? First, **vision is a versatile and informational interface that complements language**. Dense information like depth and segmentation cannot be described easily through language [Fu et al., 2024]. In a broader perspective, humans have developed many visualization techniques that are direct, efficient, and informational. Sketchpad provides LMs the opportunity to use them. Second, in Sketchpad, multimodal LMs can **plan and reason based on the intermediate visual artifacts** they created. In contrast, in prior modular vision work [Gupta and Kembhavi, 2023; Surís et al., 2023; Yang et al., 2023a], multimodal modules follow a predefined plan by either humans or code. Sketchpad is much more flexible and robust to errors. For example, suppose object detection makes an error. The LM can (in principle) find the error by viewing the bounding boxes, and change its following plans, but prior methods cannot. Third, as discussed next, **the plans of multimodal LMs are similar to human plans**, and therefore likely benefit from the fact that the underlying LMs have seen data with similar reasoning patterns.

Do LMs have the same plans as humans? We conduct a human study on all geometry problems and 10 problems on each vision task. On geometry, humans draw the same auxiliary line as GPT-4o 80% of the time. On vision, we show 2 human subjects the full plan of GPT-4o, which they rate is valid in 92.8% of instances. Most errors are caused by failures in the vision specialists (e.g., fail to detect an object) and mistakes in simple visual question answering, rather than planning.

Model	Geometry	Maxflow	Convexity	Winner ID
LLaVA-NeXT-13B	11.1	7.8	50.39	5.8
+ oracle Sketchpad	22.2	10.2	50.0	36.7
LLaVA-NeXT-34B	26.1	0.8	81.6	49.0
+ oracle Sketchpad	28.3	14.1	87.1	49.4

Table 4.4: Open-source LLaVA models' performance on math tasks. The oracle Sketchpad uses the visual artifact generated in the last action of GPT-4o + Sketchpad as inputs.

Experiments on open-source models. Can sketches like diagrams, plots, and auxiliary lines facilitate existing open-source multimodal LMs? To answer this question, we conduct the experiments in Table 4.4. We use the visual artifacts generated in the last action of GPT-4o + Sketchpad experiment as the image input for open-source LLaVA-NEXT models [Liu et al., 2024b]. We can see that this oracle Sketchpad brings consistent improvement to math tasks and boosts mathematical reasoning.

4.6 Summary

We present Visual Sketchpad, a framework that provides multimodal LMs with the tools necessary to generate intermediate sketches to reason over tasks. For complex mathematical reasoning tasks, Sketchpad yields large performance gains, by visualizing auxiliary lines, math functions, graphs, and games during reasoning. For visual reasoning tasks, we add vision specialists to Sketchpad. The LM can call these specialists during reasoning, observing the visualization of these specialists' predictions (e.g., bounding boxes from the object detection model; masks from the segmentation model), and then conduct further planning and reasoning. Experiments show that Sketchpad enhances the LMs' performance across all tasks, and sets new state-of-the-art results. Ultimately, Sketchpad represents a step toward endowing LMs with more human-like multimodal intelligence, leveraging the complementary strengths of language and vision to tackle increasingly complex reasoning challenges.

Chapter 5

TIFA: Accurate and Interpretable Text-to-Image Faithfulness Evaluation with Question Answering

In the previous chapters, we demonstrate how to empower multimodal understanding with interactions. In this chapter, we focus on how to use interactions as a signal to train and evaluate text-to-image generation models. We present TIFA [Hu et al., 2023b], an evaluation metric for generated images that are based on interactions with multimodal LMs. TIFA is the first work that uses multimodal LMs to evaluate image generation models, and it has been adopted in many follow-up works on image generation model training and evaluation.

5.1 Introduction[1]

While we welcome artistic freedom when we commission art from artists, images produced by deep generative models [Ramesh et al., 2021; Rombach et al., 2021; Ramesh et al., 2022; Saharia et al., 2022; Yu et al., 2022] should conform closely to our requests. Despite the advances in generative models, it is still challenging for models to produce images faithful to users' intentions [Petsiuk et al., 2022; Feng et al.; Lee et al., 2023a; Liu et al., 2022b, 2023c]. For example, current models often fail to compose multiple objects [Petsiuk

[1]This work involved several collaborators. I am the primary author that proposed the idea and conducted the experiments. All authors jointly wrote the paper.

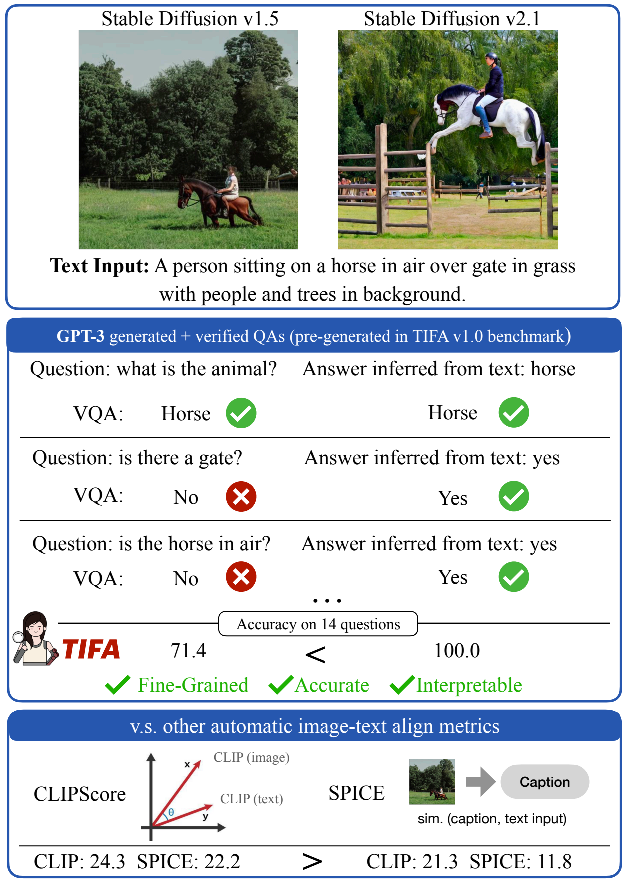


Figure 5.1: Illustration of how TIFA works, and comparison with the widely-used CLIPScore and SPICE metrics. Given the text input, TIFA uses GPT-3 to generate several question-answer pairs, and a QA model filters them (3 out of 14 questions for this text input are shown). TIFA measures whether VQA models can accurately answer these questions given the generated image. In this example, TIFA indicates that the image generated by SD v2.1 is better than that by v1.5, while CLIP and SPICE yield the opposite result.

et al., 2022; Feng et al.; Liu et al., 2022b], bind attributes to the wrong objects [Feng et al.], and struggle in generating visual text [Liu et al., 2023c]. Today, there are efforts to address these challenges: researchers are imposing linguistic structure with diffusion guidance to produce images with multiple objects [Feng et al.]; others are designing reward models trained using human feedback to better align generations with user intention [Lee et al., 2023a]. However, progress is difficult to quantify without accurate and interpretable evaluation measures that explain when and how models struggle.

A critical bottleneck, therefore, is the lack of reliable automatic evaluation metrics for text-to-image generation faithfulness. One of the popular metrics is CLIPScore [Hessel et al., 2021], which measures the cosine similarity between the CLIP embeddings [Radford et al., 2021] of the text input and the generated

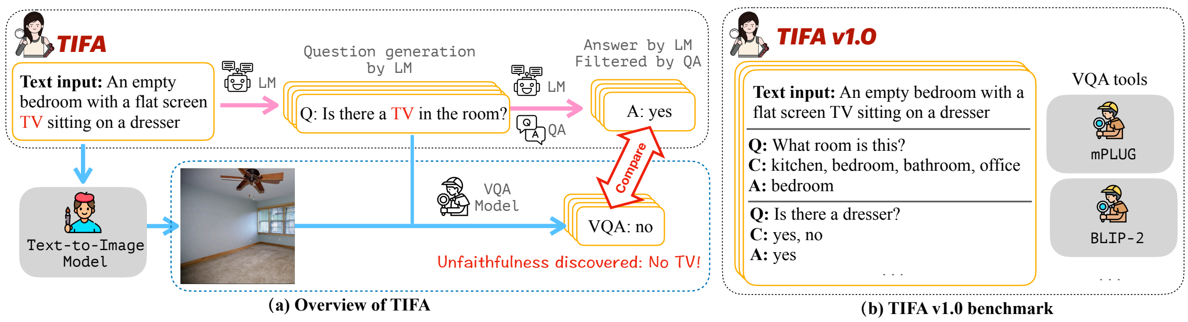


Figure 5.2: (a) Overview of how TIFA evaluates the faithfulness of a synthesized image. TIFA uses a language model (LM), a question-answering (QA) model, and a visual-question-answering (VQA) model. Given a text input, we generate several question-answer pairs with the LM and then filter them via the QA model. To evaluate the faithfulness of a synthesized image to the text input, a VQA model answers these visual questions using the image, and we check the answers for correctness. **(b) TIFA v1.0 benchmark.** While TIFA is applicable to any text prompt, to allow direct comparison across different studies, and for ease of use, we introduce the TIFA v1.0 benchmark, a repository of text inputs along with pre-generated question-answer tuples with answer choices. To evaluate a text-to-image model, a user first produces the images for the text inputs in TIFA v1.0 and then performs VQA with our provided tools on generated images to compute TIFA.

image. However, since CLIP is not effective at counting objects [Radford et al., 2021], or reasoning compositionally [Ma et al., 2023], CLIPScore is unreliable and often inaccurate. Another family of evaluation metrics uses image captions, in which an image captioning model first converts the image into text, and then the image caption is evaluated by comparing it against the text input. Unfortunately, using captioning models is insufficient since they might decide to ignore salient information in images or focus on other non-essential image regions [Kasai et al., 2022b]; for example, a captioning model might say that the images in Figure 5.1 are "a person sitting on a horse in air over gate in grass with people and trees in background". Moreover, evaluating text (caption) generation is inherently challenging [Kasai et al., 2022a; Khashabi et al., 2022]. Another recent text-to-image evaluation is DALL-Eval [Cho et al., 2023], which employs object detection to determine if the objects in the texts are in the generated images. However, this approach only works on synthesized text and measures faithfulness along the limited axes of objects, counting, colors, and spatial relationships but misses activities, geolocation, weather, time, materials, shapes, sizes, and other potential categories we often ask about when we recall images from memory [Krishna et al., 2019].

To address the above challenges, we introduce TIFA, a new metric to evaluate text-to-image generation faithfulness. Our approach is illustrated in Figure 5.2. Given a repository of text inputs, we automatically generate question-answer pairs for each text via a language model (here, GPT-3 [Brown et al., 2020]). A question-answering (QA) system (here, UnifiedQA [Khashabi et al., 2020]) is subsequently used to verify and filter these question-answer pairs. To evaluate a generated image, we use a visual-question-answering (VQA) system (here, mPLUG-large [Li et al., 2022], BLIP-2 [Li et al., 2023b], etc.) to answer the questions given the generated image. We measure the image's faithfulness to the text input as the accuracy of the answers generated by the VQA system. While the accuracy of TIFA is dependent on the accuracy of the VQA model, our experiments show that TIFA has much higher correlation with human judgments than CLIPScore (Spearman's ρ = 0.60 vs. 0.33) and captioning-based approaches (Spearman's ρ = 0.60 vs. 0.34). Additionally, since the LMs and VQA models will continue to improve, we hypothesize that TIFA will continue to be more reliable over time. Also, our metrics can automatically detect when elements are missing in the generation. For example, in Figure 5.2, TIFA detects that the generated image does not contain a TV.

To promote the use of our new evaluation metric, we release TIFA v1.0, a large-scale text-to-image generation benchmark containing 4K diverse text inputs, sampled from the MSCOCO captions [Lin et al., 2014], DrawBench [Saharia et al., 2022], PartiPrompts [Yu et al., 2022], and PaintSkill [Cho et al., 2023]. Each input comes with a pre-generated set of question-answer pairs, resulting in 25K questions covering 4.5K distinct elements. These questions have been automatically generated and pre-filtered using a question-answering model. This benchmark also comes with different VQA models [Wang et al., 2022a; Kim et al., 2021b; Wang et al., 2022b; Li et al., 2023b, 2022; Hu et al., 2023a] that can be used to evaluate generative models and can be easily extended to use future VQA models when they become available.

We conduct a comprehensive evaluation of current text-to-image models using TIFA v1.0. Thanks to TIFA's ability to detect fine-grained unfaithfulness in images, we find that current state-of-the-art models are good at rendering common objects, animals, and colors, but still struggle in composing multiple objects, reasoning about spatial relations, and binding the correct activity for each entity. In addition, our ablation experiments show that TIFA is robust to different VQA models. Future researchers can use TIFA v1.0 to compare their text-to-image models' faithfulness across different studies. Also, future generative models may focus on addressing the weaknesses of current models that TIFA discovered. In addition, with TIFA, users can customize evaluations with their own text inputs and questions [Ethayarajh and Jurafsky, 2020]; for example, a future TIFA benchmark could focus on counting or scene text.

5.2 The TIFA Metric

We introduce a framework for automatically estimating the faithfulness of an image to its text prompt. Given a text input T, we aim to measure the faithfulness of the generated image I. An overview of our metric is illustrated in Figure 5.2. From T, we generate N multiple-choice question-answer tuples $\{Q_i, C_i, A_i\}_{i=1}^N$, in which Q_i is a question, C_i is a set of answer choices, and $A_i \in C_i$ is the gold answer. The answer A_i can be inferred given T, Q_i, and C_i. Next, for each question Q_i, we use a VQA model to produce an answer $A_i^{\text{VQA}} = \max_{a \in C_i} p(a \mid I, Q_i)$. We define the faithfulness between the text T and image I as the VQA accuracy:

$$faithfulness(T, I) = \frac{1}{N} \sum_{i=1}^{N} 1[A_i^{\text{VQA}} = A_i] \tag{5.1}$$

The range of our faithfulness score is $[0, 1]$. It is maximized when we have a performant VQA model, and the image I accurately covers the information in the text T so that for any question Q, which can be answered given T can also be answered given I. Several key design decisions will be addressed in later sections: how to generate questions (§5.2.1), how to control the question quality (§5.2.2), and how to answer those questions (§5.2.3). Finally, we give a step-by-step qualitative example of TIFA in Figure 5.4.

5.2.1 Question-Answer Generation

Our main challenge is to generate diverse questions that cover all elements of the text input evenly. We also simplify the question-generation pipeline into a single GPT-3 [Brown et al., 2020] completion, so that TIFA can exploit the power of recent language models (LM) and work with updated black-box LMs (e.g., ChatGPT) in the future.

Inspired by prior work [Changpinyo et al., 2022], given a text prompt T, we generate the question-answer tuples $\{Q_i, C_i, A_i\}_{i=1}^N$ via the pipeline illustrated in Figure 5.3. Different from prior work, which relies on multiple components, our pipeline is completed by a single inference run via in-context learning with GPT-3 [Brown et al., 2020; Wei et al., 2022; Hu et al., 2022; Press et al., 2023; Su et al., 2023], thereby avoiding the need for intermediate human annotations. We annotate 15 examples and use them as in-context examples for GPT-3 to follow. Here we take the text *"A photo of three dogs."* as an example. Each in-context example contains the following steps:

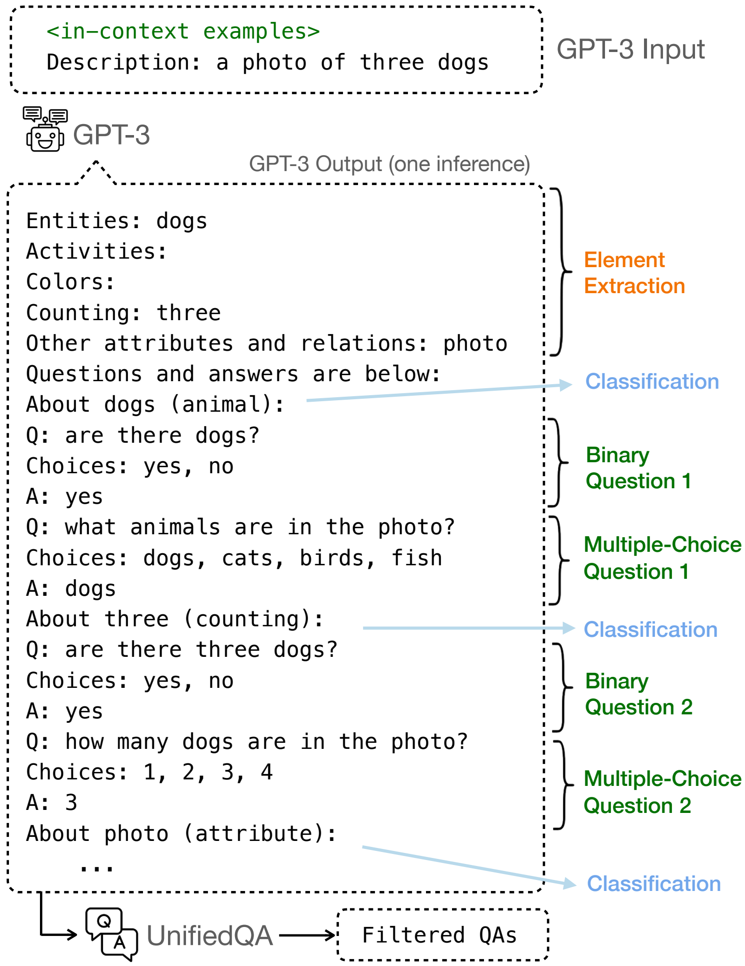


Figure 5.3: Our question-answer pair generation pipeline. The whole pipeline can be executed via a single inference of GPT-3 via in-context learning. Given the text prompt, GPT-3 first extracts the elements and then generates two questions for each element. The GPT-3 output is then parsed and filtered by UnifiedQA.

Element extraction Given text prompt T, GPT-3 will first extract all elements $\{v_i\}_{i=1}^m$ following prior work [Changpinyo et al., 2022] (for the in-context examples, we perform element extraction manually). The elements include noun phrases (including named entities), verbs, adjectives, adverbs, and parse tree spans with no more than 3 words altogether. For the above example, the elements are *photo, three, dogs*.

Element category classification For each element v_i, following [Krishna et al., 2019], we classify the elements into one of the following 12 categories: *object, activity, animal, food, counting, color, material, spatial, location, shape, attribute*, and other. As shown in Figure 5.3, text generated from GPT-3 contains the category corresponding to each question.For example, "three" is "counting", and "dogs" is classified as "animal." This step allows a detailed analysis of the text-to-image model's ability in each category.

Question generation conditioned on elements For each element v_i, we generate two questions. The first is a question that should be answered "yes" for a faithful generated image, and the second question has v_i as its answer. For example, two questions are generated for the element *"three"*. The first is "are there three dogs?", and the choices are {yes, no}. The second is "how many dogs are there?", and the choices are {1,2,3,4}. These two types of questions make our evaluations diverse and robust to surface-level differences.

Completing above steps by prompting GPT-3 once As mentioned earlier, for each text T, the whole pipeline can be completed by one GPT-3 inference. We annotated 15 in-context examples that cover all types of questions. The prompt format is shown in Figure 5.3. Our in-context examples follow the same format, and identical examples are used for all text inputs, leading to a fixed and limited amount of human annotation cost. We use *code-davinci-002* engine for question generation, and the decoding temperature is 0.

5.2.2 Question Filtering

To ensure the quality of generated images, we use UnifiedQA [Khashabi et al., 2020] to verify the GPT-3 generated question-answer pairs and filter out the ones that GPT-3 and UnifiedQA do not agree on. UnifiedQA[2] is a state-of-the-art multi-task question-answering model that can answer both multiple-choice and free-form questions. Denote the UnifiedQA model as QA. Given the text T, question Q_i, choices C_i, and answer A_i, Let $A_i^f = QA(T, Q_i)$ be the free-form answer, and $A_i^{mc} = QA(T, Q_i, C_i)$ be the multiple-choice answer. We keep the question if $A_i = A_i^{mc}$ and the word-level F_1 score between A_i^f and A_i is greater than 0.7. We conduct a human evaluation on 1000 filtered question-answer pairs. Only 7 are considered not reasonable (e.g., generated choices do not include a correct answer).

5.2.3 VQA Models

Since our questions contain a diverse set of visual elements (e.g., activity, art style), we use open-domain pre-trained vision-language models as our VQA model (rather than closed-class classification models fine-tuned

[2]Model checkpoint we use: https://huggingface.co/allenai/unifiedqa-v2-t5-large-1363200.

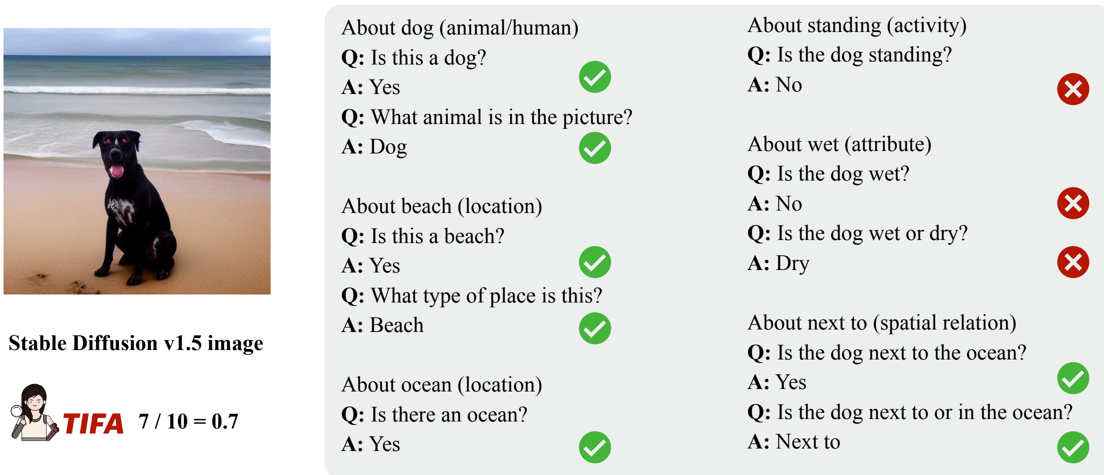


Figure 5.4: Step-by-step qualitative example of TIFA metric. Given a text input, we first generate question-answer pairs and filter them. Then we run VQA models on the generated image to get the TIFA score.

on VQAv2 [Goyal et al., 2017]). We provide tools to easily perform VQA on arbitrary images and questions, based on 5 state-of-the-art VQA models trained with distinct data and strategies.

Vision-language models The general pre-trained vision-language model are **GIT-large** [Wang et al., 2022a], **VILT-B/32** [Kim et al., 2021b], **OFA-large** [Wang et al., 2022b], and **mPLUG-large** [Li et al., 2022]. These models are pre-trained on a large amount of image-text pairs, and downstream image-to-text tasks like image captioning and visual question answering. Notice that these models have not been trained to answer multiple-choice questions. For each question, we first decode the free-form answer and then choose the choice that has the highest similarity with the decoded answer, measured by SBERT [Reimers and Gurevych, 2019]. Another model we use is **BLIP-2 FlanT5-XL** [Li et al., 2023b], in which a VIT [Dosovitskiy et al., 2021] is connected with a frozen FlanT5 [Chung et al., 2024] via a lightweight transformer. This model allows for performing multiple-choice VQA directly due to the flexibility of the LM.

Recommended VQA model Based on considerations over the accuracy, correlation with human judgments, and run time, as of March 2023, we suggest using **mPLUG-large** as the VQA model for TIFA. Analysis is given in Section 5.4.4. Like the LM and QA components, the VQA component can be updated in the future as the technology improves.

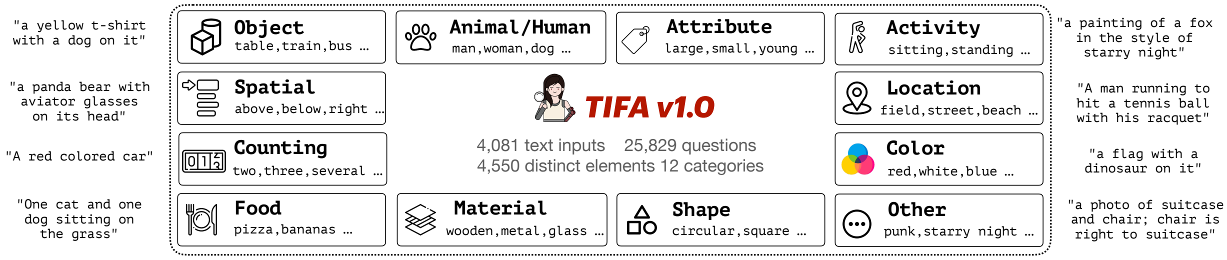


Figure 5.5: Statistics and diversity of TIFA v1.0. The text inputs contain elements from 12 categories (e.g., object, spatial, and counting). We show the most common elements from each category. In addtion, we also show some example text inputs on the sides.

5.3 TIFA v1.0: Benchmark for Text-to-Image Generation Faithfulness

In this section, we introduce TIFA v1.0, a text-to-image generation faithfulness benchmark based on the evaluation method discussed in Section 5.2. The benchmark consists of 4,081 diverse text inputs paired with 25,829 question-answer pairs. Each question is classified into one of the categories discussed in Section 5.2.1. The benchmark also comes with Python pip-installable APIs to perform VQA with various state-of-the-art VQA models on arbitrary visual questions. The overall TIFA for each text-to-image model is computed by averaging TIFA scores of images generated from each text input in the benchmark.

5.3.1 Text Collections

We collect 4,081 text inputs to benchmark text-to-image models' generation ability on diverse tasks. 2,000 text inputs are image captions from **COCO** validation set [Lin et al., 2014]. These captions have corresponding gold images. Since text-to-image models are often used to create abstract art, we also collect 2,081 text inputs from previous works that do not correspond to any real image, referred to as "free from." All text inputs we use contain ≥ 3 words. We include 161 from **DrawBench** used in Imagen [Saharia et al., 2022] (texts that are categorized as "misspellings" and "rare words" are removed); 1420 from **PartiPrompt** used in Parti [Yu et al., 2022] (texts in category "abstract" are removed); and 500 texts from **PaintSkill** used in DALL-Eval [Cho et al., 2023].

Statistics	
# of prompts	4,081
- # of COCO captions	2,000
- # of free form (DrawBench, PartiPrompt, PaintSkill prompts)	2,081
# of questions	25,829
- # of binary questions	17,226
- # of multiple-choice questions	8,603
avg. # of questions per prompt	6.3
avg. # of words per prompt	10.5
avg. # of elements per prompt	4.3

Table 5.1: Statistics of TIFA v1.0.

5.3.2 Statistics and Diversity

Table 5.1 shows the basic statistics of the TIFA v1.0 benchmark. We illustrate TIFA v1.0's diversity in Figure 5.5. TIFA v1.0 contains questions about 4,550 distinct elements, which are categorized into 12 categories. The number of times each type of element occurs in the text input is object (7,854), animal/human (3,501), attribute (3,399), activity (2,851), spatial (2,265), location (1,840), color (1,743), counting (986), food (911), material (209), shape (69), and other (201). The "other" category includes notions often used in abstract art, such as “starry night” and “steampunk.” The accuracy of VQA on a particular genre measures the text-to-image model's ability in the corresponding aspect.

5.3.3 Finetuned Open-Source Language Model for Question Generation

For TIFA v1.0, we use GPT-3 to generate the questions. While benchmarking with TIFA v1.0 is a deterministic process, using TIFA to create a new benchmark might not be deterministic as the underlying question generator (GPT-3 in our case) might change privately. To promote deterministic benchmark generation, we fine-tune and release a LLaMA 2 (7B) [Touvron et al., 2023] model that parses the captions and generates questions for arbitrary texts, using TIFA v1.0 questions as training examples.[3]

[3] LLaMA 2 question generation model checkpoint:
https://huggingface.co/tifa-benchmark/llama2_tifa_question_generation

5.4 Experiments

In this section, we first show that TIFA has substantially higher correlations with human judgments than prior metrics on text-to-image faithfulness (§5.4.1). Then we present a comprehensive evaluation of existing text-to-image models using TIFA v1.0 (§5.4.2), highlighting the challenges of current text-to-image models (§5.4.3). Finally, we conduct an analysis of TIFA's robustness against different VQA models (§5.4.4). For all experiments, we use **mPLUG** as the VQA model for TIFA unless stated otherwise. The models we evaluate include AttnGAN [Xu et al., 2017], X-LXMERT [Cho et al., 2020], VQ-Diffusion [Gu et al., 2021], minDALL-E [Kim et al., 2021a], and Stable Diffusion v1.1, v1.5, and v2.1 [Rombach et al., 2021].

5.4.1 Correlation with Human Judgements

To compare TIFA with prior evaluation metrics, we first conduct human evaluations of the text-to-image models on the 1-5 Likert scale on text-to-image faithfulness. Then we compare TIFA with other metrics based on their correlation with human judgments.

Likert scale on text-to-image faithfulness Annotators are asked to answer on a scale of 1 (worst) to 5 (best) to the question *"Does the image match the text?"*. The detailed annotation guidelines are in Appendix B.1. Annotators are asked to focus on text-to-image faithfulness rather than image quality. The Likert scale should be based on how many elements in the text prompt are missed or misrepresented in the image. Objects are more important than attributes, relations, and activities. If an object is missed in the image, then all related attributes, activities, relations, etc. are also considered lost. An example is given in Figure 5.6.

We collect annotations of 800 generated images on 160 text inputs from TIFA v1.0. For each prompt, we sample an image from the 5 most recent generative models we evaluated, i.e., minDALL-E, VQ-Diffusion, Stable Diffusion v1.1, v1.5, and v2.1. We collect 2 annotations per image and average over the scores as the single "faithfulness" score. The inter-annotator agreement measured by Krippendorf's α is 0.67, indicating "substantial" agreement.

Baselines We compare our evaluation with two families of reference-free metrics on text-image match introduced in Section 2.2. The first is the **caption-based method**. We use the state-of-the-art **BLIP-2**

Figure 5.6: Illustration of our Likert scale annotation guideline. Annotators are asked to give a score of 1 to 5 based on how many elements in the text prompt are missed or misrepresented in the image. The missed elements are underlined.

FlanT5-XL [Li et al., 2023b] as the captioning model. The second approach is **CLIPScore** [Hessel et al., 2021; Radford et al., 2021]. We use CLIP (VIT-B/32) [Radford et al., 2021] to compute the score.

	Spearman's ρ	Kendall's τ
Caption-Based		
BLEU-4	18.3	18.8
ROUGE-L	32.9	24.5
METEOR	34.0	27.4
SPICE	32.8	23.2
CLIPScore	33.2	23.1
Ours		
TIFA (VILT)	49.3	38.2
TIFA (OFA)	49.6	37.2
TIFA (GIT)	54.5	42.6
TIFA (BLIP-2)	55.9	43.6
TIFA (mPLUG)	**59.7**	**47.2**

Table 5.2: Correlations between each evaluation metric and human judgment on text-to-image faithfulness, measured by Spearman's ρ and Kendall's τ.

TIFA has a much higher correlation with human judgments than prior metrics. The correlations between each evaluation metric and human judgment are shown in Table 5.2. For caption-based evaluations, we use metrics BLEU-4 [Papineni et al., 2002], ROUGE-L [Lin, 2004], METEOR [Banerjee and Lavie, 2005], and SPICE [Anderson et al., 2016]. TIFA has higher correlations with human judgments than all previous evaluation metrics on all VQA models. TIFA (mPLUG) yields the highest correlation with human judgments among all VQA models.

5.4.2 Benchmarking Text-to-Image Models

Figure 5.7 shows the average TIFA score text-to-image models get on TIFA v1.0. The detailed scores with each VQA model on each element type are provided in Table 5.3. We can see a clear trend of how text-to-image models evolve over time. There is a jump in TIFA score after DALL-E [Ramesh et al., 2021] is released, from about 60% to 75%.

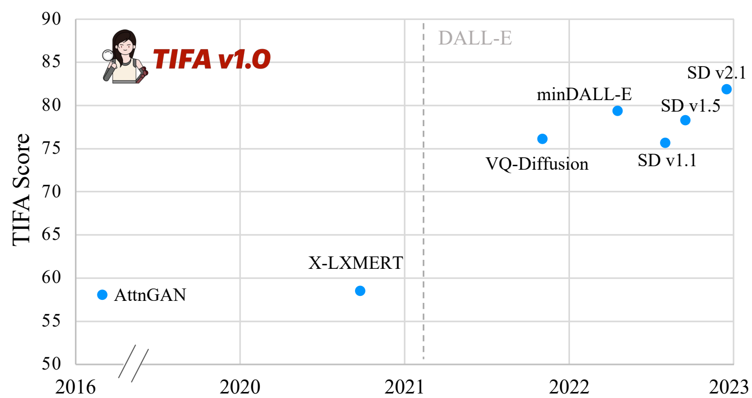


Figure 5.7: Average TIFA score of text-to-image models on the TIFA v1.0 benchmark. The horizontal axis shows their release dates.

	VQA Accuracy by Element Category							
	shape	other	counting	spatial	attribute	activity	food	object
AttnGANXu et al. [2017]	42.0	47.8	41.9	70.8	53.6	64.3	48.1	56.3
X-LXMERTCho et al. [2020]	34.8	46.8	41.7	70.9	55.2	65.4	52.4	57.0
Stable Diffusion v1.1Rombach et al. [2021]	66.7	68.7	66.0	69.4	74.6	73.8	79.7	75.1
VQ-DiffusionGu et al. [2021]	63.8	64.2	61.6	73.5	75.4	76.7	80.0	74.2
Stable Diffusion v1.5Rombach et al. [2021]	65.2	72.1	66.6	72.9	78.0	76.9	81.4	78.4
minDALL-EKim et al. [2021a]	**69.6**	**74.6**	69.0	74.7	77.0	79.5	**82.8**	79.9
Stable Diffusion v2.1Rombach et al. [2021]	66.7	72.1	**73.3**	**76.1**	**78.8**	**82.0**	82.2	**82.4**

					TIFA by text source		**Overall**
	location	color	animal/human	material	COCO	free-form	
AttnGANXu et al. [2017]	60.4	56.5	58.6	61.7	67.5	47.4	58.1
X-LXMERTCho et al. [2020]	69.1	54.8	52.8	61.2	68.1	47.7	58.6
Stable Diffusion v1.1Rombach et al. [2021]	78.4	75.7	78.2	80.4	79.3	72.2	75.7
VQ-DiffusionGu et al. [2021]	77.9	**84.2**	79.0	80.9	79.8	72.6	76.2
Stable Diffusion v1.5Rombach et al. [2021]	79.9	78.8	80.6	84.7	81.9	74.9	78.4
minDALL-EKim et al. [2021a]	82.1	83.7	78.9	86.1	83.5	75.5	79.4
Stable Diffusion v2.1Rombach et al. [2021]	**82.8**	83.6	**85.2**	**88.5**	**86.3**	**77.7**	**82.0**

Table 5.3: Detailed evaluation of each text-to-image model on TIFA v1.0.

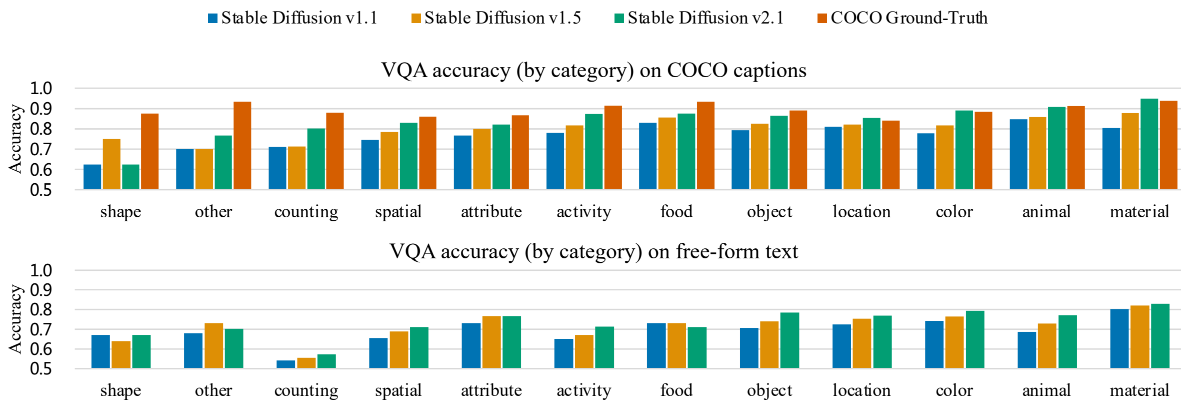


Figure 5.8: Accuracy on each type of question in the TIFA v1.0 benchmark. The text-to-image models are Stable Diffusion v1.1, v1.5, and v2.1. We order the categories by the average score Stable Diffusion v2.1 gets on corresponding questions. For COCO captions, we also include the accuracy of the ground-truth images for reference.

5.4.3 Findings on Current Text-to-Image Models

Figure 5.8 shows accuracy on each type of question in TIFA v1.0 for Stable Diffusion v1.1, v1.5, and v2.1. The score on each type reflects the text-to-image models' faithfulness in each type of visual element. To the best of our knowledge, TIFA is the only automatic evaluation method that can provide such a detailed fine-grained analysis of image generation. We separate the scores on COCO captions and other text inputs. For COCO captions, we also include the accuracy on the ground-truth images for reference. We summarize our findings in the following paragraphs.

Generating images from captions vs. free-form text From Figure 5.8, we can see that VQA accuracy is higher on the COCO captions than on other text inputs. We hypothesize that COCO captions correspond to real images, while other text inputs may correspond to compositions that cannot be found in real-world photos (e.g. "a blue apple").

What elements are text-to-image models struggling with? Based on the scores of each category in Figure 5.8, we can see that Stable Diffusion models are performing well on material, animal/human, color, and location in terms of text-to-image faithfulness. However, they yield low accuracy on questions involving **shapes**, **counting**, and **spatial relations**. "Other" mainly contains **abstract art notions**, and models are also struggling with them. There is also a big gap between the synthesized images and real images on the

COCO captions. Future work can explore various directions (e.g., training data/loss and model architecture) to improve text-to-image models' faithfulness in these aspects.

Why are ground-truth images not getting perfect scores? Ground-truth images in COCO do not get perfect scores because 1) the COCO captions contain a substantial amount of noise from crowd workers [Kasai et al., 2022b] and 2) VQA models are not perfect. Real images have higher accuracy in all categories except material, color and location, where differences are small. It is left to future work to determine whether this is simply due to noise or it is an area where assessment can be improved.

Stable Diffusion is evolving. We can see the consistent trend that Stable Diffusion models are improving in their later versions in most of the element categories. The exceptions are "shape" for both prompt sources, "other" and "food" for the free-form text inputs without gold images.

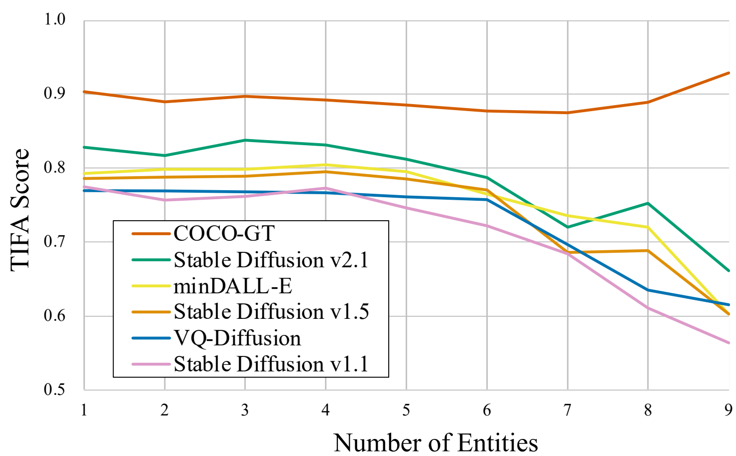


Figure 5.9: TIFA vs. numbers of entities (objects, animals/humans, and food) in the text input. The accuracy starts to drop when more than 5 entities are added to the text, showing that compositionality is hard for text-to-image models. Meanwhile, TIFA scores for COCO ground-truth (GT) images remain consistent.

Composing multiple objects is challenging. Figure 5.9 shows how the number of entities (objects, animals/humans, food) in the text input affects the average TIFA score. When there are more than 5 entities, The TIFA score starts to drop rapidly for all text-to-image models, consistent with similar findings in other vision-language evaluations [Grunde-McLaughlin et al., 2021; Gandhi et al., 2022]. For reference, we also add the real images in COCO in this figure. The TIFA score on real images is rather consistent and does not change as the number of entities increases. This quantitatively shows that composing multiple objects is challenging for current text-to-image models. One possible reason is that the CLIP text embedding, which is used to train Stable Diffusion, lacks compositionality, as investigated in [Ma et al., 2023].

5.4.4 Analysis of VQA Models

One major concern of TIFA is that VQA models can introduce some errors. Table 5.2 shows that TIFA has a much higher correlation with human judgment than the previous metrics, regardless of the choice of the VQA models; here we conduct a more detailed analysis.

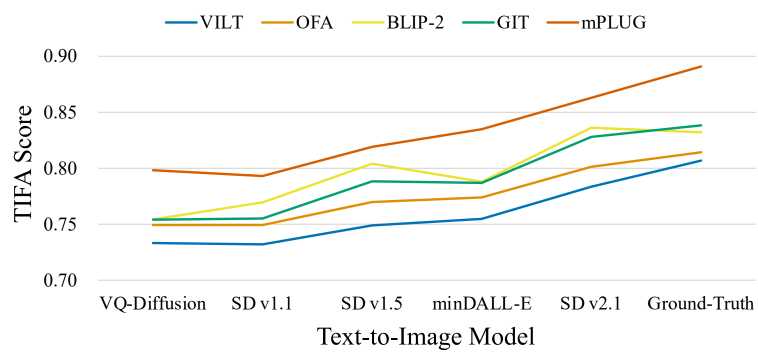


Figure 5.10: Several text-to-image models' TIFA score on COCO captions, measured by different VQA models. We also include the accuracy of ground-truth images for reference.

Sensitivity of TIFA to VQA models Figure 5.10 shows several recent text-to-image models' TIFA scores on the COCO captions in TIFA v1.0, measured by different VQA models. We also include the TIFA scores on the ground-truth COCO images for reference. TIFA scores computed by different VQA models show a similar trend on these text-to-image models. Also, the ground-truth images get the highest TIFA score. We also computed Spearman's ρ of TIFA scores given by different VQA models. The pairwise correlation between all VQA models is greater than 0.6.

Humans performing VQA To conduct further analysis on the VQA models, we ask annotators to answer the multiple-choice visual questions in TIFA v1.0. These annotations help us evaluate the accuracy of each VQA model. For multiple-choice questions, we add the option "None of the above" for human evaluation.

We collect annotations of 1029 questions on 126 images. Each question is answered by two annotators.

The inter-annotator agreement measured by Krippendorf's α is 0.88. A third annotator is involved if two annotators disagree, and the final answer is chosen by the majority vote.

	VILT	OFA	GIT	BLIP-2	mPLUG
VQA Acc.	76.1	77.1	79.1	81.0	84.5
TIFA Corr.	60.9	63.7	72.5	75.6	76.8

Table 5.4: Comparison of VQA models. The first row is the VQA accuracy, using the human VQA answers as reference. The second row is Spearman's correlation between TIFA scores calculated by each VQA model and the human VQA.

Which VQA model should we use? Table 5.4 reports the accuracy of each VQA model and the correlation between TIFA scores calculated by VQA model answers and human answers. We observe that higher model performance is directly related to the TIFA score's correlation with human judgments. **mPLUG** has the highest accuracy.

Another important factor to consider is the runtime. We measure the inference speed of each VQA model on NVIDIA A40 GPU with batch size 1 over the Stable Diffusion v2.1 images (768×768 pixels). For one question, VILT takes 0.08s on average; OFA, GIT, and mPLUG all take about 0.25s; BLIP-2 takes 0.73s. Based on the above results, we choose **mPLUG** as the default VQA model for TIFA v1.0 because it is the most accurate while being reasonably fast.

Separation of Text-to-Image Errors and VQA Errors Suppose an image gets a wrong answer given a visual question. Then the image generation or the VQA model might have made an error. Based on the human VQA results, we separate these two kinds of errors in Figure 5.11. If human VQA gives the wrong answer, then we suspect the generated image has an error. Otherwise, the image is correct but the VQA model is making an error. Figure 5.11 shows that the majority of errors are made by the text-to-image models. For mPLUG, less than 25% errors are due to the VQA model. This suggests that the TIFA framework is a viable evaluation method despite its inherent challenges.

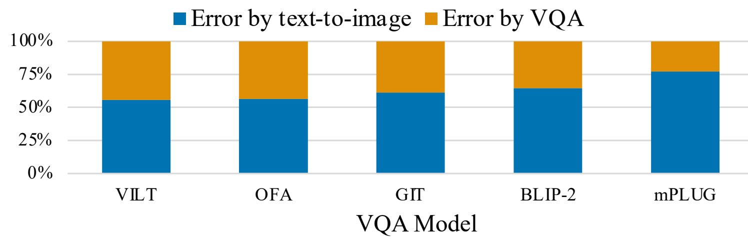


Figure 5.11: Source of the error when VQA gets the wrong answer.

5.5 Training Text-to-Image Generation with TIFA

TIFA opens the door for text-to-image models to gain training signals through interactions with multimodal LMs. There are many follow-ups on improving image generation with TIFA. For example, [Karthik et al., 2023] uses TIFA to re-rank the model-generated images. [Singh and Zheng, 2023] uses TIFA as iterative feedback to reweight prompts in text-to-image generation models.

Following these works, we introduce DreamSync [Sun et al., 2025], a paradigm to fine-tune text-to-image models with TIFA. DreamSync is demonstrated in Figure 5.12. DreamSync is inspired by the rejection sampling works on LLMs [Touvron et al., 2023]. We first collect a diverse set of synthetic prompts. For each prompt, we sample many images from the text-to-image model, and use TIFA to select the ones that are most faithful to the text input. We also uses an aesthetics model VILA [Ke et al., 2023] as an extra dimension for image selection. Finally, we finetune the text-to-image model with the selected image. Experiments show that this approach can interatively improve text-to-image model's faithfulness to text, while preserving its visual appeal.

5.6 Summary

We present TIFA, a new automatic text-to-image faithfulness evaluation metric using VQA. Compared with prior metrics, TIFA is fine-grained, interpretable, and better aligned with human judgments. Based on this metric, we introduce the TIFA v1.0, a large-scale text-to-image benchmark containing 4K prompts and 25K questions. We conduct a comprehensive study of current text-to-image models using TIFA v1.0 and highlight the limitations of current generative models. We quantitatively show that current image generation models still struggle in counting, spatial relations, and composing multiple objects. Finally, we conduct extensive

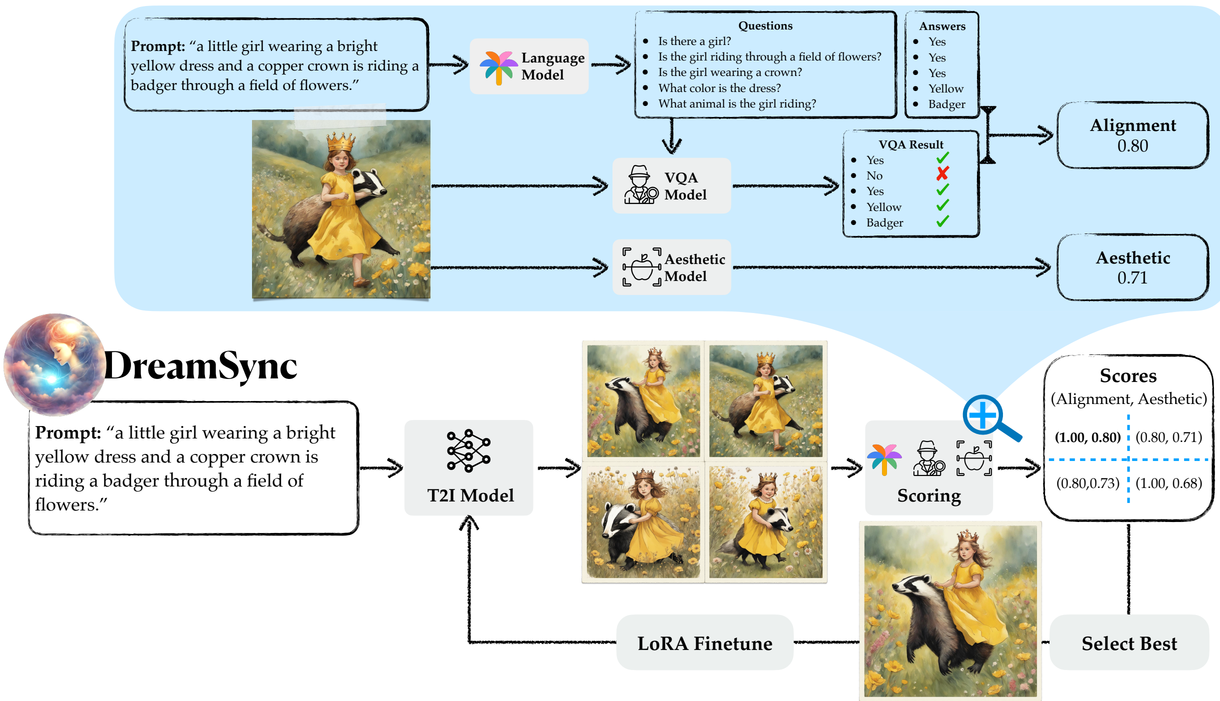


Figure 5.12: DreamSync. Given a prompt, a text-to-image generation model generates multiple candidate images, which are evaluated by two models. We use TIFA to compute the alignment between the image and the text, and use VILA to score the model's visual appeal. The best images chosen are collected to fine-tune the text-to-image model. This process can repeat indefinitely until convergence on feedback is achieved.

analysis and human evaluation, demonstrating that TIFA is robust to different VQA models. We hope TIFA will help evaluate future work on image generation and become increasingly sophisticated as it is upgraded with new LM, QA, and VQA components. We also show that TIFA works well as a training signal for image generation models.

Chapter 6

NavigationBench: Dialogue-Level Evaluation of LM Agents in a Multi-turn Interactive Environment

6.1 Introduction

Recent advances in language models have significantly expanded their capabilities. However, two key challenges remain in improving their real-world helpfulness. First, most AI models operate in isolation, lacking access to external data, environments, or tools—severely limiting their ability to interact with the real world. In response, several organizations have proposed protocols to bridge this gap. For example, OpenAI introduced function calling [OpenAI, 2023a], and Claude has introduced the Model Context Protocol (MCP) [Anthropic, 2024a]. To better understand the strengths and limitations of such integrations, we need controlled and open experimental settings that evaluate model interaction with different tool types.

Second, current AI systems are typically designed to directly answer user queries, rather than collaborate interactively. Yet in real-world use cases, user goals are often underspecified and require back-and-forth dialogue to uncover preferences and needs. While there has been substantial progress in interactive AI research, we still lack an evaluation framework that enables dialogue-level comparisons of system behavior. Existing open benchmarks generally fall into one of three categories. First, some evaluate system responses against one or more reference dialogues. Although this approach supports automatic evaluation, it fails to capture the diversity of valid dialogue trajectories and limits assessment of end-to-end performance. Second, other benchmarks rely on interactions with real users, such as in the Amazon Alexa Prize competition [Venkatesh et al., 2018] or CMU's Let's Go system [Raux et al., 2005]. While this method supports realistic evaluation, it is costly, and a large number of user interactions are needed to capture the variability of different users interacting with different systems. Moreover, the risk of system failures may render it unsuitable for certain tasks.

To address these issues, we propose a flexible evaluation framework compatible with both simulated and real users. Recent progress in LLMs enables the creation of high-quality simulated users, making it feasible to support open research on dialogue agents at scale. Our framework consists of two key components: (1) a task paired with an interactive environment, where solutions can be ranked by humans, and (2) a simulated user that interacts with the AI system. In each test case, the AI system engages in a multi-turn dialogue with the simulated user to solve the task, and its output is evaluated based on a task-specific scoring function. This structure allows for scalable and rigorous assessment across a broad range of real-world tasks.

To demonstrate this framework, we introduce NavigationBench, a multi-turn dialogue benchmark focused on navigation tasks that require dynamic user interaction. NavigationBench includes a large set of simulated map environments, task scenarios validated by human annotators, and simulated users powered by state-of-the-art language models, enabling automatic end-to-end evaluation. Crucially, as language models improve, the simulator can be upgraded accordingly. The framework also supports human-in-the-loop evaluation when needed, enabling direct comparison between simulated and real user interactions.

In the following sections, we detail our evaluation framework (§6.2), introduce NavigationBench (§6.3), and present experimental results from applying this task to a range of AI systems (§6.5).

6.2 Evaluation Framework

This section details our evaluation framework, designed for assessing AI systems in a wide range of problem-solving tasks grounded in interactive environments. The framework enables end-to-end, dialogue-level comparison of AI systems, provided that human annotators can meaningfully rank solutions based on a scenario. It comprises the following components:

1. Task. The first component is a problem-solving task that requires AI systems to collaborate with users. These tasks are often paired with interactive environments—such as coding platforms, web search tools, databases, or customized MCP servers—that the assistant can interact with. Each test case is defined by a scenario that specifies both user requirements and preferences. These scenarios guide the behavior of the simulated user. Human annotators rank the solutions of the system based on these scenarios.

2. Simulated User. Since it is expensiveve to have real humans talk with all the assistant systems, to support dialogue-level evaluation, we design one or more simulated users per task. For each test case, the simulated user adheres to the scenario and emulates natural user behavior—avoiding overly verbose or unnaturally informative turns, and instead engaging in incremental, clarifying interaction. It responds in context-sensitive ways, mirroring how real users behave when their goals are partially specified. Importantly, the simulated user design allows for flexibility and future upgrades as LLMs continue to improve.

3. Evaluation Metric. To avoid the high cost of human annotators observing each dialogue between the system and the simulated user, we score the assistant by the final solution it gives. For each scenario, we have human annotators rank possible solutions, and this ranking will be used to evaluate all the dialogues. We evaluate the AI system based on the Mean Reciprocal Rank (MRR) of the final solution it arrives at through interaction with the simulated user. Because human rankings can vary across annotators, we use the average Reciprocal Rank to aggregate the preferences:

$$RR_i = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{R_{ij}} \tag{6.1}$$

where R_{ij} is the rank of the system's solution according to annotator j ($R_{ij} = \infty$ if not ranked), and N is the number of annotators. RR_i ranges from 0 to 1. A perfect consensus that ranks a solution first gets 1.

Let M be the number of dialogues in the test set. The MRR is computed as

$$MRR = \frac{1}{M} \sum_{i=1}^{M} RR_i \tag{6.2}$$

6.3 NavigationBench

To demonstrate our framework, we introduce NavigationBench, a multi-turn dialogue benchmark that simulates user–assistant interaction in a vehicle navigation setting. Each dialogue simulates a scenario in which the user follows a pre-defined route and makes an unplanned stop (e.g., for food, fuel, or rest). The assistant helps refine options through interactive dialogue, akin to using a smart navigation app with conversational capabilities.

1. Environment. NavigationBench provides a simulated map environment similar to commercial navigation apps. The language model can query this environment through Python functions or custom protocols to search for nearby stops, retrieve detailed location data (e.g., reviews, prices), and estimate travel times and detours.

2. Scenarios. Scenarios are designed to elicit meaningful dialogue. Each includes the user's requirements, preferences, contextual background, and initial request. We generate scenarios using GPT-4.1-mini, following a bootstrap procedure inspired by self-instruct [Wang et al., 2022c]. Scenarios span four stop types: restaurants, fuel stations, hotels, and rest areas. Human annotators refined the GPT-generated scenarios for quality and engagement, and filtered out the ones that does not make sense. This process yields 450 scenarios, including 150 for the development set and 300 for the testing set.For each scenario, three human annotators rank only the valid stops (those satisfying all requirements) based on how well they match user preferences. Below is an example scenario:

- **User requirement:** Breakfast diner within 20 minutes' drive.
- **User preference:** Wants a classic American diner serving hearty breakfast options like pancakes and omelets. Prefers good customer service and quick serving time. Budget around $20 per person. Interested in diners with a nostalgic feel and positive reviews.
- **Context:** The user started the trip early in the morning and needs a substantial breakfast before continuing the drive. There are 5 people in total in the car.
- **Time:** 8:30AM

- **User initial request:** Find a breakfast diner close by.

For each scenario, we have 3 human annotators rank possible stops on the map. The human annotators only rank the stops that satisfy all user requirements. The stops are ranked by how much they satisfy the users' preferences.

3. Simulated User. The simulated user is implemented using GPT-4.1-mini with a task-specific system prompt. The prompt includes the scenario details and behavioral instructions. Full prompt content is shown in §C.1.

4. Evaluation Metrics. Human annotators often vary in their rankings of candidate stops. For instance, given the same scenario, three annotators may each prefer a different gas station. To account for this variability, we evaluate assistant performance using the Mean Reciprocal Rank (MRR), as described in §6.2, based on how its selected solution compares to the rankings of all three annotators. To estimate the upper bound of human agreement, we compute a human MRR by averaging the MRR of each individual annotator's top choice against the rankings of the other two.

6.4 Baseline Assistant

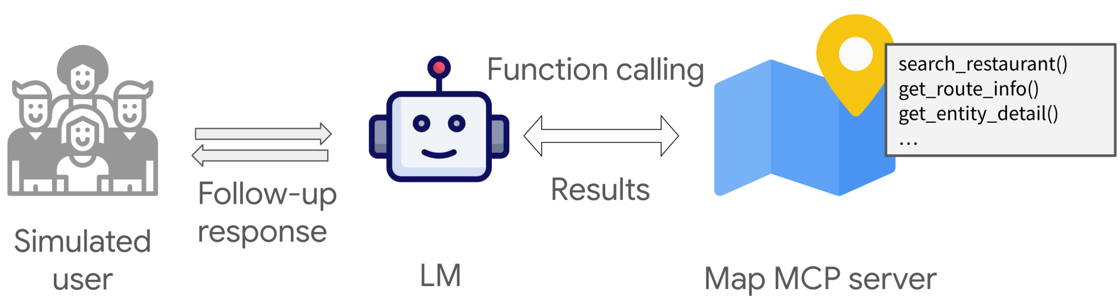


Figure 6.1: Illustration of the baseline agent we built for NavigationBench.

We implement a baseline agent that uses the LLM's function-calling abilities, as shown in Figure 6.1. The system comprises a language model and a map MCP server. The language model serves as the assistant, managing dialogue with both the user and the environment.

Dialogue Flow. After each user turn, the assistant decides whether to query the MCP server. The server returns structured JSON responses. The assistant can issue further queries to the MCP server or generate a user-facing response, including clarification questions or suggested stops. The assistant model has full access to the conversation history, including user messages, function calls, server responses, and its own replies.

Dialogue Termination. When the assistant calls a special function, $add\_stop(entity\_id)$, the stop is added to the route, and the dialogue ends. The chosen entity is then evaluated according to the human rankings.

Details of the MCP server. The MCP server has the following functions that the LM can call.

- **Search function**. This function takes a dictionary that specifies search criteria as inputs, and returns a list of entities on the map that satisfy the criteria. The criteria contain the entity type ($e.g.$, restaurant), time and distance constraints ($e.g.$, within 10 minutes off the current route), and entity constraints ($e.g.$, with rating higher than 4.0, Italian cuisine, including "homemade pasta" in their offerings, and budget smaller than $ 50). One issue is that there might be too many entities that satisfy all the constraints. In that case, we return a list of a maximum of 20 entities, together with the total count of entities that meet all criteria.

- **Get entity detail.** This function takes the unique ID of the entity as input, and returns its detailed information. For example, for a restaurant, it will return the cuisine, price range, rating, features, offerings, and a short summary of comments. In addition, it will also give the temporal and spatial information, including the coordinate of the entity, how much time/distance it takes to arrive at the entity, and the off-route time/distance.

- **Get route information.** The function takes two coordinates as inputs, and returns the time and distance needed to travel between these two coordinates.

6.5 Experiments

6.5.1 Experiment settings

We use GPT-4.1-mini as the simulated user. We use a large range of latest LMs in the baseline agent to test their agentic abilities, including open-weight models like Llama-3.1-8B, Llama-3.3-70B [Dubey et al., 2024], Qwen2.5-7B, Qwen3-14B, Qwen3-32B, Qwen3-235B-A22B [Yang et al., 2025], and proprietary models including Gemini-2.0-Flash, Gemini-2.5-Flash [Team et al., 2023], GPT-4o, GPT-4.1-mini, and GPT-4.1 [OpenAI, 2023b]. We limit the maximum number of turns to 15. Because LMs (both the simulated user and the assistant) can generate different responses that might lead to different solutions, we run each scenario four times.

6.5.2 Results

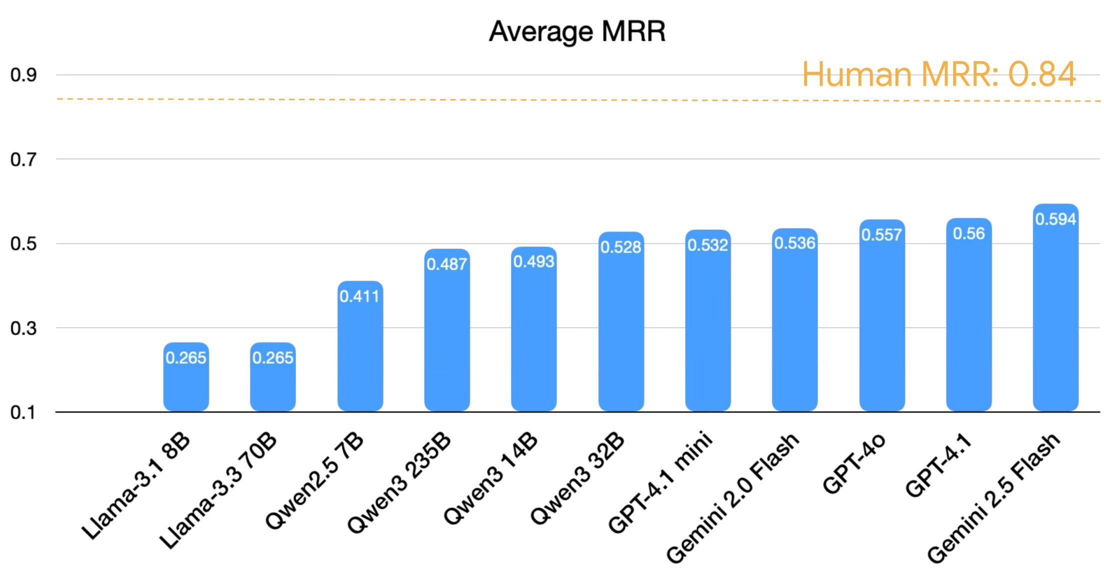


Figure 6.2: Test set performance of the baseline agent using different LMs, which is measured by the MRR of the solution. For reference, MRR between 3 human annotators is 0.84.

Figure 6.2 shows the test set MRR for each model. Human annotator MRR is 0.838, while models range from 0.265 to 0.594. Gemini-2.5-Flash performs best, though all models exhibit a gap from human-level quality. Proprietary models outperform open-weight ones.

Figure 6.3 breaks down performance by stop type for Gemini-2.5-Flash and LLaMA-3.1-8B. We also

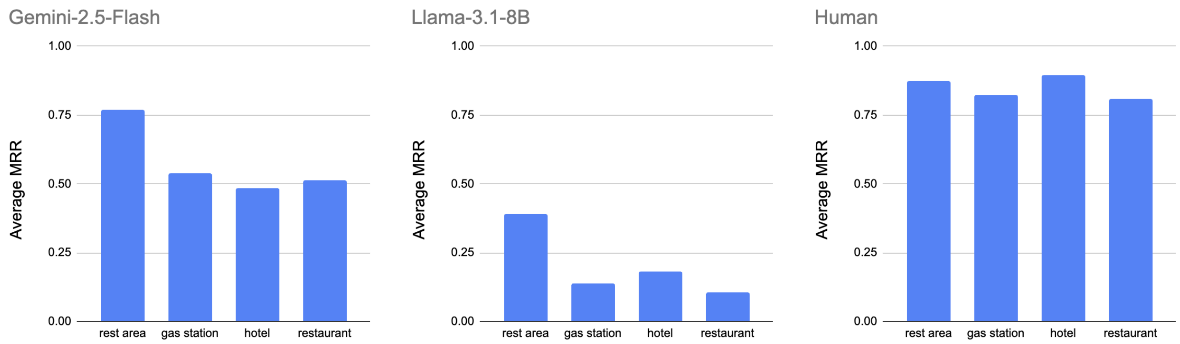


Figure 6.3: Performance of Gemini-2.5-Flash, Llama-3.1-8B, and human annotators, on different types of scenarios in the development set.

include the human MRR for reference. Both LM assistants perform best on rest areas. A possible explanation is that there are fewer attributes for rest areas. Tasks involving gas stations, hotels, and restaurants may be more challenging due to denser or more nuanced choice spaces. We can also see that the AI assistant has a bigger gap compared with humans in these entity types.

6.6 Analysis

Beyond solution quality, other user experience factors matter. For instance, fewer turns and more concise assistant responses are generally preferable—especially for users under high cognition load, for example, when driving.

Model	Avg. MRR (↑)	Avg. #Turns (↓)	Avg. Assistant Token/Turn (↓)
Llama-3.1-8B	0.265	8.47	53.7
Llama-3.3-70B	0.265	10.16	47.4
Qwen2.5-7B	0.411	6.94	85.7
Qwen3-235B-A22B	0.487	6.50	53.7
Qwen3-14B	0.493	6.91	40.5
Qwen3-32B	0.528	6.63	56.0
GPT-4.1-mini	0.532	6.21	49.2
Gemini-2.0-Flash	0.536	5.20	**40.1**
GPT-4o	0.557	5.42	54.6
GPT-4.1	0.560	**4.70**	44.8
Gemini-2.5-Flash	**0.594**	4.99	50.6

Table 6.1: The MRR, average number of assistant turns per dialogue, and the average number of tokens per assistant turn, for each assistant model on the test set.

We present more dialogue statistics in Table 6.1. Here we compute the average number of turns of each dialogue, and the average number of assistant output tokens per turn. Here the assistant output token refers to the number of tokens the assistant says to the user. We can see that Gemini-2.5-Flash, while getting the highest MRR, is relatively concise in terms of the number of turns and the number of assistant tokens. In general, a model with stronger agentic capability can give users a better solution in fewer turns.

	DSTC2	M2M	MultiWOZ-2.0	NavigationBench	
				Gemini-2.5-Flash	GPT-4.1
Avg. turns per dialogue	14.49	9.86	13.46	10.00	9.37
Avg. words per turn	8.54	8.24	13.13	27.69	24.58

Table 6.2: Dialogue-level statistics across four corpora. The NavigationBench and M2M have a simulated user, whereas other datasets have a human user.

We compare the dialogues of NavigationBench with other dialogue datasets in Table 6.2, including DSTC 2 [Williams et al., 2013], M2M [Shah et al., 2018], and MultiWOZ-2.0 [Budzianowski et al., 2018]. The table shows the average number of turns per dialogue, counting user and assistant turns separately. In addition, it shows the average number of words per turn, averaging over both user and assistant turns. While other benchmarks involve a static set of dialogues, our benchmark is dynamic, and the statistics can change with different simulated users and assistant LMs, and GPT-4.1-mini as the user LLM. Here we use the two best-performing LMs, Gemini-2.5-Flash and GPT-4.1, as the assistant LMs. NavigationBench exhibits significantly more words per assistant turn, leading to a higher overall average. The longer assistant turns reflect more complex behaviors of the assistant. The assistant may present multiple options for the user, *e.g.*, “I found 11 Italian restaurants that serve pizza and pasta, the first one is [...], the second one is [...], Do any of those sound good, or would you like to hear about other options?” The assistant may ask the user to relax some criteria when it coundn’t find a suitable one, *e.g.*, “I can’t find any restaurants that fit those criteria. It seems that "oysters" and "water view" are a difficult combination in this area. Would you be open to considering a restaurant with just oysters, even without a water view? Or a restaurant with a water view that doesn’t serve oysters?”

6.7 Summary and Discussion

We present a modular evaluation framework for assessing AI assistants through end-to-end, dialogue-level interaction. We also introduce NavigationBench as a case study, simulating realistic navigation dialogues where assistants must identify suitable stops based on both explicit and latent user preferences. The framework supports scalable evaluation using simulated users.

Our experiments across a diverse range of language models reveal a clear performance gap between current systems and human-level capability, highlighting the need for further research in dialogue planning, tool use, and preference understanding.

A key strength of our framework lies in its generalizability. It can be easily adapted to other domains—such as coding, data analysis, or planning—as long as solutions can be meaningfully ranked. Moreover, each component is modular: the task, environment, assistant model, and simulated user can be swapped out independently. The simulated user can be updated as technology further evolves. This flexibility enables broad applicability in benchmarking emerging capabilities and system behaviors across different tool-use and user-alignment tasks.

Chapter 7

Conclusion

7.1 Summary and Contributions

In this thesis, we explore solutions to key challenges in building AI systems with more sophisticated multimodal understanding, generation, and interaction. We introduce multimodal interactive intelligence to address these challenges. We enable multimodal interactions into the model reasoning process, incorporate interaction into multimodal generation evaluation, and develop an evaluation framework that is better suited to interactive problem-solving systems.

7.1.1 Enhancing multimodal understanding with interactions

We introduce BLINK, a benchmark focusing on core visual perception abilities that are often overlooked in existing multimodal LM evaluations. BLINK comprises 14 tasks that, while simple for humans, pose significant challenges for current multimodal LMs, revealing that these models can "see" but often do not truly "perceive." BLINK has been widely used in technical reports of latest multimodal LMs, including Gemini [Team et al., 2023], GPT [OpenAI, 2023b], Seed-1.5-VL [Guo et al., 2025].

To address the identified perception gaps, we proposed Visual Sketchpad, a framework that empowers multimodal LMs to generate intermediate visual artifacts (sketches) as part of their reasoning process. This approach, inspired by how humans use sketching to aid problem-solving, allows LMs to interact with specialized vision tools and perform step-by-step multimodal reasoning, significantly improving performance on diverse mathematical and visual tasks.

7.1.2 Evaluating and training image generation models via interactions

We develop TIFA (Text-to-Image Faithfulness evaluation with question Answering), a novel reference-free metric that leverages multimodal LMs to assess the faithfulness of generated images to their text prompts. TIFA automatically generates questions about the input text and uses a multimodal LM to determine if these questions can be correctly answered using the generated image. Alongside the metrics, we also introduce the TIFA benchmark, featuring over 4,000 prompts and 25,000 questions. Experiments demonstrated that TIFA has a significantly higher correlation with human judgments compared to previous metrics and provides interpretable, fine-grained feedback on model performance across various visual elements. TIFA is the first work that uses multimodal LM to evaluate model-generated images. This approach has become the status quo approach for evaluating the alignment of multimodal contents with text, and is widely adopted in the latest multimodal generation benchmarks [Cho et al.; Wiles et al., 2024; Lin et al., 2024]. TIFA can also serve as an effective training signal to improve text-image alignment in generation models. For example, DreamSync [Sun et al., 2025] shows that rejection sampling with TIFA effectively improves text-to-image models' faithfulness to user input.

7.1.3 Dialogue-level evaluation framework for interactive systems

We present a modular evaluation framework for assessing interactive systems with direct dialogue-level comparison. The key components are a simulated user that the AI assistant can interact with, and the evaluation metric that captures human preferences for different solutions. We also introduce NavigationBench as a case study for this framework .This benchmark simulates dynamic dialogues where users require assistance for unplanned stops during a journey, necessitating collaborative problem-solving. Our experiments on NavigationBench show that there is still a clear performance gap between existing systems and human-level capability, highlighting the need for future research on interactive AI systems.

7.1.4 Conclusion

The studies in this thesis provide new tools and insights for developing AI systems that can more effectively understand, generate, and interact across multiple modalities. The proposed benchmarks and evaluation methods aim to guide future research by highlighting current limitations and providing robust ways to measure progress. We have released all code and data for the work discussed in this thesis to support future research on building advanced multimodal AI systems.

7.2 Future directions

Building on the work presented in this thesis, several promising directions for future research emerge to further enhance multimodal interactive intelligence.

7.2.1 Follow-up directions for works in this thesis

Training better models with Sketchpad. Sketchpad focuses on existing off-the-shelf LMs. Future work may explore the training side of Sketchpad. For example, recent multimodal models like Unified-IO 2 [Lu et al., 2024a] and Chameleon [Team, 2024] are natively multimodal and can output both text and images. Sketchpad may emerge as a new paradigm for instruction tuning these models. Also, we can use reinforcement learning to enable models to learn when and how to sketch more effectively.

Applying Sketchpad in more areas. Sketchpad can be applied in more areas. For example, for robotics, we can apply Sketchpad to search for small things in a crowded space, highlight the object of interest, and zoom the camera for a better view or use depth estimation to help navigation.

Extending NavigationBench In this thesis, we experiment with different assistant LMs for NavigationBench. This benchmark can be further extended in many different directions. For example, it is interesting to dive into the simulated user, see how the user behavior affects the dialogue outcome, and how to train an LM that better simulates real users. Also, future research can enrich the task with more types of entities and more challenging scenarios, for example, finding stops with requirements and preferences involving multiple entities.

7.2.2 More general directions

Improving visual perception of multimodal LMs. Research could also focus on improving the underlying visual perception of multimodal LLMs directly through other ways. For example, future work can leverage the insights from BLINK to guide pre-training and fine-tuning objectives and data curation.

Scaling and refining interaction-based evaluation and training. TIFA has shown promise in evaluating and training image generation models through interaction-based approaches. Future research could expand this paradigm to other generative tasks, such as audio, video, or 3D asset generation. Future work can also explore how to more efficiently use such interaction-based signals for model training, perhaps through more advanced reinforcement learning techniques or by integrating human feedback more directly.

Developing more sophisticated environment and interactive agents. Our proposed evaluation framework provides a platform for studying grounded multi-turn interactions. Future work can build upon this by creating even more complex and diverse interactive environments and tasks. Enhancing the capabilities of simulated users to exhibit a wider range of behaviors and adapt more dynamically to different AI assistants would be valuable. Moreover, future work can use these environments to develop agents with better long-term memory, seamlessly switch between different tools, and with better planning capabilities to handle complex sequences of actions and reasoning steps.

Exploring self-training in complex multimodal environments. Our ultimate goal is to build AI systems that can efficiently self-improve by interacting with the multimodal world and learning from feedback, just like humans do. This involves better model architectures, interaction paradigms, world simulators, training objectives, and algorithms. The studies in this thesis are all building blocks towards this goal. For instance, AI agents could be trained in simulated environments like NavigationBench, using self-generated visual aids akin to Visual Sketchpad to navigate and solve problems, with their performance and learning guided by interaction-based reward signals similar to those in TIFA. This could lead to a virtuous cycle in which agents continuously refine their understanding, generation, and interaction capabilities through active exploration and learning within rich multimodal contexts. Developing self-training methodologies that can efficiently leverage these components in unison represents a significant and exciting challenge for future research.

Bibliography

Manoj Acharya, Kushal Kafle, and Christopher Kanan. 2019. Tallyqa: answering complex counting questions. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 8076–8084.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *Proceedings of the European Conference on Computer Vision*, pages 382–398. Springer.

Anthropic. 2024a. Introducing the Model Context Protocol. `https://www.anthropic.com/news/model-context-protocol`.

Anthropic. 2024b. Introducing the next generation of Claude. `https://www.anthropic.com/news/claude-3-family`.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2425–2433.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. 2023. OpenFlamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. 2017. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Harry Barrow, J Tenenbaum, A Hanson, and E Riseman. 1978. Recovering intrinsic scene characteristics. *Comput. Vis. Syst*, 2(3-26):2.

Sean Bell, Kavita Bala, and Noah Snavely. 2014. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33:1 – 12.

William Berrios, Gautam Mittal, Tristan Thrush, Douwe Kiela, and Amanpreet Singh. 2023. Towards language models that can see: Computer vision through the lens of natural language. *arXiv preprint arXiv:2306.16410*.

Michael J Black and Padmanabhan Anandan. 1993. A framework for the robust estimation of optical flow. In *1993 (4th) International Conference on Computer Vision*, pages 231–236. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Soravit Changpinyo, Doron Kukliansky, Idan Szpektor, Xi Chen, Nan Ding, and Radu Soricut. 2022. All you may need for VQA are image captions. In *North American Chapter of the Association for Computational Linguistics*.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023a. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023b. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*.

Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. 2016. Single-image depth perception in the wild. *Advances in Neural Information Processing Systems*, 29.

Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. 2023c. PaLI-X: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Jaemin Cho, Yushi Hu, Jason Michael Baldridge, Roopal Garg, Peter Anderson, Ranjay Krishna, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian Scene Graph: Improving reliability in fine-grained evaluation for text-to-image generation. In *The Twelfth International Conference on Learning Representations*.

Jaemin Cho, Jiasen Lu, Dustin Schwenk, Hannaneh Hajishirzi, and Aniruddha Kembhavi. 2020. X-LXMERT: Paint, caption and answer questions with multi-modal transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8785–8805.

Jaemin Cho, Abhay Zala, and Mohit Bansal. 2023. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3043–3054.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

OpenCompass Contributors. 2023a. Opencompass: A universal evaluation platform for foundation models. `https://github.com/open-compass/opencompass`.

XTuner Contributors. 2023b. Xtuner: A toolkit for efficiently fine-tuning llm. `https://github.com/InternLM/xtuner`.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023a. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*.

Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. 2023b. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. 2024. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. 2024. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of NLP leaderboards. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. 2023. Eva: Exploring the limits of masked visual representation learning at scale. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 19358–19369.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Reddy Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *The Eleventh International Conference on Learning Representations*.

Stanislav Frolov, Tobias Hinz, Federico Raue, Jörn Hees, and Andreas R. Dengel. 2021. Adversarial text-to-image synthesis: A review. *Neural networks : the official journal of the International Neural Network Society*, 144:187–209.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. 2023a. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Deqing Fu, Ruohao Guo, Ghazal Khalighinejad, Ollie Liu, Bhuwan Dhingra, Dani Yogatama, Robin Jia, and Willie Neiswanger. Isobench: Benchmarking multimodal foundation models on isomorphic representations. In *First Conference on Language Modeling*.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. 2023b. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *Advances in Neural Information Processing Systems*, 36:50742–50768.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024. Blink: Multimodal large language models can see but not perceive. In *Proceedings of the European Conference on Computer Vision*, pages 148–166. Springer.

Xingyu Fu, Sheng Zhang, Gukyeong Kwon, Pramuditha Perera, Henghui Zhu, Yuhao Zhang, Alexander Hanbo Li, William Yang Wang, Zhiguo Wang, Vittorio Castelli, Patrick Ng, Dan Roth, and Bing Xiang. 2023c. Generate then select: Open-ended visual question answering guided by world knowledge. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2333–2346, Toronto, Canada. Association for Computational Linguistics.

Xingyu Fu, Ben Zhou, Ishaan Chandratreya, Carl Vondrick, and Dan Roth. 2022. There's a time and place for reasoning beyond the image. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1138–1149, Dublin, Ireland. Association for Computational Linguistics.

Xingyu Fu, Ben Zhou, Sihao Chen, Mark Yatskar, and Dan Roth. 2023d. Interpretable by design visual question answering. *arXiv preprint arXiv:2305.14882*.

Mona Gandhi, Mustafa Omer Gul, Eva Prakash, Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. 2022. Measuring compositional consistency for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5046–5055.

Vinod Goel. 1995. *Sketches of thought*. MIT press.

Gabriela Goldschmidt. 1991. The dialectics of sketching. *Creativity research journal*, 4(2):123–143.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. 2021. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11287–11297.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. 2021. Vector quantized diffusion model for text-to-image synthesis. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10686–10696.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. 2025. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*.

Agrim Gupta, Piotr Dollar, and Ross Girshick. 2019. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962.

Chris Harris, Mike Stephens, et al. 1988. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer.

Richard Hartley and Andrew Zisserman. 2003. *Multiple view geometry in computer vision*. Cambridge university press.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Joseph Le Bras, and Yejin Choi. 2021. Clipscore: A reference-free evaluation metric for image captioning. In *Conference on Empirical Methods in Natural Language Processing*.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and

Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Cheng-Yu Hsieh, Si-An Chen, Chun-Liang Li, Yasuhisa Fujii, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. 2023. Tool documentation enables zero-shot tool-usage with large language models. *arXiv preprint arXiv:2308.00675*.

Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A. Smith, and Jiebo Luo. 2023a. PromptCap: Prompt-guided image captioning for VQA with GPT-3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2963–2973.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-Context Learning for Few-Shot Dialogue State Tracking. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 2627–2643.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. 2023b. TIFA: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20406–20417.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. 2024a. Visual Sketchpad: Sketching as a visual chain of thought for multimodal language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. 2024b. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9590–9601.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. 2024. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*.

Shyamgopal Karthik, Karsten Roth, Massimiliano Mancini, and Zeynep Akata. 2023. If at first you don't succeed, try, try again: Faithful diffusion-based text-to-image generation by selection. *arXiv preprint arXiv:2305.13308*.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Lavinia Dunagan, Jacob Morrison, Alexander Fabbri, Yejin Choi, and Noah A. Smith. 2022a. Bidimensional leaderboards: Generate and evaluate language hand in hand. In *North American Chapter of the Association for Computational Linguistics*.

Jungo Kasai, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Daniel Morrison, Ronan Le Bras, Yejin Choi, and Noah A. Smith. 2022b. Transparent human evaluation for image captioning. In *North American Chapter of the Association for Computational Linguistics*.

Junjie Ke, Keren Ye, Jiahui Yu, Yonghui Wu, Peyman Milanfar, and Feng Yang. 2023. Vila: Learning image aesthetics from user comments with vision-language pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10041–10051.

Apoorv Khandelwal, Ellie Pavlick, and Chen Sun. 2023. Analyzing modular approaches for visual question decomposition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2590–2603.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 1896–1907, Online. Association for Computational Linguistics.

Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A. Smith, and Daniel Weld. 2022. GENIE: Toward reproducible and standardized human evaluation for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Saehoon Kim, Sanghun Cho, Chiheon Kim, Doyup Lee, and Woonhyuk Baek. 2021a. mindall-e on conceptual captions.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021b. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*.

Ranjay Krishna, Michael S. Bernstein, and Li Fei-Fei. 2019. Information maximizing visual question generation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2008–2018.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73.

Zihang Lai, Senthil Purushwalkam, and Abhinav Gupta. 2021. The functional correspondence problem. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15772–15781.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. 2023a. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*.

Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *CHI Conference on Human Factors in Computing Systems*, CHI '22. ACM.

Mina Lee, Megha Srivastava, Amelia Hardy, John Thickstun, Esin Durmus, Ashwin Paranjape, Ines Gerard-Ursin, Xiang Lisa Li, Faisal Ladhak, Frieda Rong, Rose E Wang, Minae Kwon, Joon Sung Park, Hancheng Cao, Tony Lee, Rishi Bommasani, Michael S. Bernstein, and Percy Liang. 2023b. Evaluating human-language model interaction. *Transactions on Machine Learning Research*.

Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. 2024a. Evaluating and improving compositional text-to-visual generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5290–5301.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024b. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13299–13308.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023a. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*.

Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, He Chen, Guohai Xu, Zheng Cao, Ji Zhang, Songfang Huang, Fei Huang, Jingren Zhou, and Luo Si. 2022. mPLUG: Effective and efficient vision-language learning by cross-modal skip-connections. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7241–7259, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, Lei Zhang, and Jianfeng Gao. 2024c. Segment and recognize anything at any granularity. In *European Conference on Computer Vision*, pages 467–484. Springer.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. 2024. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022a. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fangyu Liu, Guy Emerson, and Nigel Collier. 2023a. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. LLaVA-NeXT: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. 2022b. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pages 423–439. Springer.

Rosanne Liu, Dan Garrette, Chitwan Saharia, William Chan, Adam Roberts, Sharan Narang, Irina Blok, Rj Mical, Mohammad Norouzi, and Noah Constant. 2023c. Character-aware models improve visual text rendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16270–16297, Toronto, Canada. Association for Computational Linguistics.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. 2024c. Llava-plus: Learning to use tools for creating multimodal agents. In *European Conference on Computer Vision*, pages 126–142. Springer.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. 2024d. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2024e. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. 2024f. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102.

David G Lowe. 1999. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee.

Bo-Ru Lu, Nikita Haduong, Chia-Hsuan Lee, Zeqiu Wu, Hao Cheng, Paul Koester, Jean Utke, Tao Yu, Noah A Smith, and Mari Ostendorf. Does collaborative human–lm dialogue generation help information extraction from human–human dialogues? In *First Conference on Language Modeling*.

Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. 2024a. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024b. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *The 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. 2023. Crepe: Can vision-language foundation models reason compositionally? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10910–10921.

Zixian Ma, Weikai Huang, Jieyu Zhang, Tanmay Gupta, and Ranjay Krishna. 2024. m & m's: A benchmark to evaluate tool-use for m ulti-step m ulti-modal tasks. In *European Conference on Computer Vision*, pages 18–34. Springer.

David Marr. 2010. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press.

David Marr and Tomaso Poggio. 1976. Cooperative computation of stereo disparity: A cooperative algorithm is derived for extracting disparity information from stereo image pairs. *Science*, 194(4262):283–287.

Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. 2019. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv prepreint arXiv:1908.10543*.

Marvin Minsky and Seymour Papert. 1969. An introduction to computational geometry. *Cambridge tiass., HIT*, 479(480):104.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. 2024. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. In *International Conference on Machine Learning*, pages 37321–37341. PMLR.

OpenAI. 2023a. Function Calling. https://platform.openai.com/docs/guides/function-calling?api-mode=responses.

OpenAI. 2023b. Gpt-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. 2021. Benchmark for compositional text-to-image synthesis. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Vitali Petsiuk, Alexander E Siemenn, Saisamrit Surbehera, Zad Chin, Keith Tyser, Gregory Hunter, Arvind Raghavan, Yann Hicke, Bryan A Plummer, Ori Kerret, et al. 2022. Human evaluation of text-to-image models on a multi-task benchmark. *arXiv preprint arXiv:2211.12112*.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. SDXL: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. PMLR.

Antoine Raux, Brian Langner, Dan Bohus, Alan W. Black, and Maxine Eskenazi. 2005. Let's go public! taking a spoken dialog system to the real world. In *Interspeech 2005*, pages 885–888.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 29.

Ayush Sarkar, Hanlin Mai, Amitabh Mahapatra, Svetlana Lazebnik, and Anand Bhattad. 2024. Shadows don't lie and lines can't bend! generative models don't know projective geometry... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28140–28149.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. Replug: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8364–8377.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. 2023. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11987–11997.

Jaskirat Singh and Liang Zheng. 2023. Divide, evaluate, and refine: Evaluating and improving text-to-image alignment with iterative vqa feedback. *Advances in Neural Information Processing Systems*, 36:70799–70811.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. 2021. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931.

Jiao Sun, Deqing Fu, Yushi Hu, Su Wang, Royi Rassin, Da-Cheng Juan, Dana Alon, Charles Herrmann, Sjoerd Van Steenkiste, Ranjay Krishna, and Cyrus Rashtchian. 2025. DreamSync: Aligning text-to-image generation with image understanding feedback. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*

(Volume 1: Long Papers), pages 5920–5945, Albuquerque, New Mexico. Association for Computational Linguistics.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. 2023. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. RUBER: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 722–729, New Orleans, Louisiana, USA. AAAI Press.

Holly A Taylor and Barbara Tversky. 1992a. Descriptions and depictions of environments. *Memory & cognition*, 20:483–496.

Holly A Taylor and Barbara Tversky. 1992b. Spatial mental models derived from survey and route descriptions. *Journal of Memory and language*, 31(2):261–292.

Chameleon Team. 2024. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. https://github.com/InternLM/InternLM.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578.

Antonio Torralba and Aude Oliva. 2002. Depth estimation from image structure. *IEEE Transactions on pattern analysis and machine intelligence*, 24(9):1226–1238.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Barbara Tversky and Masaki Suwa. 2009. Thinking with sketches.

Barbara Tversky, Masaki Suwa, Maneesh Agrawala, Julie Heiser, Chris Stolte, Pat Hanrahan, Doantam Phan, Jeff Klingner, Marie-Paule Daniel, Paul Lee, et al. 2003. Sketches for design and design of sketches. *Human Behaviour in Design: Individuals, Teams, Tools*, pages 79–86.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Behnam Hedayatnia, Angeliki Metallinou, Rahul Goel, Shaohua Yang, and Anirudh Raju. 2018. On evaluating and comparing open domain dialog systems. In *Proceedings of the NeurIPS 2017 Workshop on Conversational AI*. Analyzes Alexa Prize socialbot logs; introduces user satisfaction and engagement metrics (turns, duration) as system-level evaluation criteria.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022a. GIT: A generative image-to-text transformer for vision and language. *Transactions on Machine Learning Research*.

John YA Wang and Edward H Adelson. 1993. Layered representation for motion analysis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–366. IEEE.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022b. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International conference on machine learning*, pages 23318–23340. PMLR.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Keqin Chen, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2024a. Cogvlm: Visual expert for pretrained language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 121475–121499. Curran Associates, Inc.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. 2024b. MINT: Evaluating LLMs in multi-turn interaction with tools and language feedback. In *The Twelfth International Conference on Learning Representations*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022c. Self-instruct: Aligning language models with self-generated instructions.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Olivia Wiles, Chuhan Zhang, Isabela Albuquerque, Ivana Kajić, Su Wang, Emanuele Bugliarello, Yasumasa Onoe, Pinelopi Papalampidi, Ira Ktena, Chris Knutsen, et al. 2024. Revisiting text-to-image evaluation with gecko: On metrics, prompts, and human ratings. *arXiv preprint arXiv:2404.16820*.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France. Association for Computational Linguistics.

Penghao Wu and Saining Xie. 2024. V?: Guided visual search as a core mechanism in multimodal llms. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13084–13094.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2017. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1316–1324.

An Yan, Zhengyuan Yang, Junda Wu, Wanrong Zhu, Jianwei Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Julian McAuley, Jianfeng Gao, et al. List items one by one: A new data source and learning paradigm for multimodal llms. In *First Conference on Language Modeling*.

An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, et al. 2023. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. *arXiv preprint arXiv:2311.07562*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023a. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. 2024a. Intercode: Standardizing and benchmarking interactive coding with execution feedback. *Advances in Neural Information Processing Systems*, 36.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024b. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 2022. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3081–3089.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023b. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023c. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Richard James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-Tau Yih. 2023. Retrieval-augmented multimodal language modeling. In *International Conference on Machine Learning*, pages 39755–39769. PMLR.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. 2022. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*. Featured Certification.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2024. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *International Conference on Machine Learning*, pages 57730–57754. PMLR.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2024. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Yanjie Ze and Xiaolong Wang. 2022. Category-level 6d object pose estimation in the wild: A semi-supervised learning approach and a new dataset. *Advances in Neural Information Processing Systems*, 35:27469–27483.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731.

Andy Zeng, Maria Attarian, brian ichter, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. 2023. Socratic models: Composing zero-shot multimodal reasoning with language. In *The Eleventh International Conference on Learning Representations*.

Jieyu Zhang, Ranjay Krishna, Ahmed Hassan Awadallah, and Chi Wang. 2024a. Ecoassistant: Using LLM assistants more affordably and accurately. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Zhuosheng Zhang, Aston Zhang, Mu Li, hai zhao, George Karypis, and Alex Smola. 2024b. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Chapter A

Appendix for Visual Sketchpad: Sketching as a Visual Chain of Thought for Multimodal Language Models

A.1 Prompts

Prompts for computer vision tasks. Here we provide the running log of GPT-4o + Sketchpad tackling a visual search problem from V^*Bench Wu and Xie [2024]. We use the same prompt template for all computer vision problems in this work. For visualization purposes, we present the prompts that contain codes differently. In our implementation, they are all text inputs to the LLM.

SYSTEM PROMPT

You are a helpful multimodal AI assistant. [MORE INSTRUCTIONS ...]

For each turn, you should first do a "THOUGHT", based on the images and text you see. If you think you get the answer to the intial user request, you can reply with "ANSWER: <your answer>" and ends with "TERMINATE".

Initial Prompt + Request

```
Here are some tools that can help you. All are python codes. They are in tools.py and will be imported for you.
The images has their own coordinate system. The upper left corner of the image is the origin (0, 0). All coordinates are normalized, i.e., the range is [0, 1].
```

```
All bounding boxes are in the format of [x, y, w, h], which is a python list. x is the horizontal coordinate of the upper-left corner of the box, y is the vertical coordinate of that corner, w is the box width, and h is the box height.
Notice that you, as an AI assistant, is not good at locating things and describe them with coordinate. You can use tools to generate bounding boxes.
You are also not good at answering questions about small visual details in the image. You can use tools to zoom in on the image to see the details. Below are the tools in tools.py:
```python
class AnnotatedImage:
    # A class to represent an annotated image. It contains the annotated image and the original image.

    def __init__(self, annotated_image: Image.Image, original_image: Image.Image=None):
        self.annotated_image = annotated_image
        self.original_image = original_image

def detection(image, objects):
    """Object detection using Grounding DINO model. It returns the annotated image and the bounding boxes of the detected objects.
    The text can be simple noun, or simple phrase (e.g., 'bus', 'red car'). Cannot be too hard or the model will break.
    The detector is not perfect, it may wrongly detect objects or miss some objects.
    Also, notice that the bounding box label might be out of the image boundary.
    You should use the output as a reference, not as a ground truth.
    When answering questions about the image, you should double-check the detected objects.

    Args:
        image (PIL.Image.Image): the input image
        objects (List[str]): a list of objects to detect. Each object should be a simple noun or a simple phrase. Should not be hard or abstract concepts like "text" or "number".

    Returns:
        output_image (AnnotatedImage): the original image, annotated with bounding boxes. Each box is labeled with the detected object, and an index.
        processed boxes (List): listthe bounding boxes of the detected objects

    Example:
        image = Image.open("sample_img.jpg")
        output_image, boxes = detection(image, ["bus"])
        display(output_image.annotated_image)
        print(boxes) # [[0.24, 0.21, 0.3, 0.4], [0.6, 0.3, 0.2, 0.3]]
        # you need to double-check the detected objects. Some objects may be missed or wrongly detected.
    """

def sliding_window_detection(image, objects):
    """Use this when you are searching for objects in the image, but the objects are not detected by the object detection model.
    In that case, the most common reason is that the object is too small such that both the vision-language model and the object detection model fail to detect it.
    This function tries to detect the object by sliding window search.
    With the help of the detection model, it tries to detect the object in the zoomed-in patches.
    The function returns a list of annotated images that may contain at leas one of the objects, annotated with bounding boxes.
    It also returns a list of a list of bounding boxes of the detected objects.

    Args:
        image (PIL.Image.Image): the input image
        objects (List[str]): a list of objects to detect. Each object should be a simple noun or a simple phrase. Should not be hard or abstract concepts like "text" or "number".

    Returns:
        possible_patches (List[AnnotatedImage]): a list of annotated zoomed-in images that may contain the object, annotated with bounding boxes.
        possible_boxes (List[List[List[Float]]]): For each image in possible_patches, a list of bounding boxes of the detected objects.
            The coordinates are w.r.t. each zoomed-in image. The order of the boxes is the same as the order of the images in possible_patches.
```
```

```python

    Example:
        image = Image.open("sample_img.jpg")
        possible_patches , possible_boxes = search_object_and_zoom(image, [" bird ", " sign "])
        for i , patch in enumerate( possible_patches ):
            print ( f"Patch {i }:")
            display ( patch .annotated_image)

        # print the bounding boxes of the detected objects in the first patch
        print ( possible_boxes [0]) # [[0.24, 0.21, 0.3, 0.4], [0.6, 0.3, 0.2, 0.3]]
    """

def segment_and_mark(image, anno_mode:list = [ 'Mask', 'Mark']):
    """Use a segmentation model to segment the image, and add colorful masks on the segmented objects . Each segment is also labeled with a number.
    The annotated image is returned along with the bounding boxes of the segmented objects .
    This tool may help you to better reason about the relationship between objects , which can be useful for spatial reasoning etc .
    DO NOT use this tool to search or detect an object . It is likely the object is small and segmentaiton does not help .
    Segmentation and marking can also be helpful for 3D and video reasoning . For example, helping you to see more clearly and analyzes the relationship between different frames of a video .

    Args:
        image (PIL.Image.Image): the input image
        anno_mode ( list , optional ): What annotation is added on the input image. Mask is the colorful masks. And mark is the number labels . Defaults to [' Mask', 'Mark'].

    Returns:
        output_image (AnnotatedImage): the original image annotated with colorful masks and number labels . Each mask is labeled with a number. The number label starts at 1.
        bboxes ( List ): listthe bounding boxes of the masks.The order of the boxes is the same as the order of the number labels .

    Example:
        User request : I want to find a seat close to windows, where should I sit ?
        Code:
        ``` python
        image = Image.open("sample_img.jpg")
        output_image, bboxes = segment_and_mark(image)
        display (output_image.annotated_image)
        ```
        Model reply: You can sit on the chair numbered as 5, which is close to the window.
        User: Give me the bounding box of that chair .
        Code:
        ``` python
        print (bboxes[4]) # [0.24, 0.21, 0.3, 0.4]
        ```
        Model reply: The bounding box of the chair numbered as 5 is [0.24, 0.21, 0.3, 0.4].
    """

def depth(image):
    """Depth estimation using DepthAnything model. It returns the depth map of the input image.
    A colormap is used to represent the depth . It uses Inferno colormap. The closer the object , the warmer the color .
    This tool may help you to better reason about the spatial relationship , like which object is closer to the camera.

    Args:
        image (PIL.Image.Image): the input image

    Returns:
        output_image (PIL.Image.Image): the depth map of the input image
```

````

    Example:
        image = Image.open("sample_img.jpg")
        output_image = depth(image)
        display (output_image)
    """

def zoom_in_image_by_bbox(image, box, padding=0.05):
    """A simple wrapper function to crop the image based on the bounding box.
    When you want to answer question about visual details in a bounding box annotated by the detection tool , you would like to zoom in on the object using this function .

    Args:
        image (PIL.Image.Image): the input image
        box ( List [ float ]) : the bounding box in the format of [x, y, w, h]
        padding ( float , optional ) : The padding for the image crop, outside of the bounding box. Defaults to 0.1. The zoom factor cannot be too small. Minimum is 0.05

    Returns:
        cropped_img (PIL.Image.Image): the cropped image

    Example:
        image = Image.open("sample_img.jpg")
        annotated_img, boxes = detection (image, "bus")
        cropped_img = zoom_in_image_by_bbox(image, boxes[0], padding=0.05)
        display (cropped_img)
    """

def overlay_images(background_img, overlay_img, alpha=0.3, bounding_box=[0, 0, 1, 1]):
    """
    Overlay an image onto another image with transparency .
    This is particularly useful visualizing heatmap while preserving some info from the original image.
    For example, you can overlay a segmented image on a heatmap to better understand the spatial relationship between objects .
    It will also help seeing the labels , circles on the original image that may not be visible on the heatmap.

    Args:
    background_img_pil (PIL.Image.Image): The background image in PIL format.
    overlay_img_pil (PIL.Image.Image): The image to overlay in PIL format.
    alpha ( float ): Transparency of the overlay image.
    bounding_box (List [ float ]) : The bounding box of the overlay image. The format is [x, y, w, h]. The coordinates are normalized to the background image. Defaults to [0, 0, 1, 1].

    Returns:
    PIL.Image.Image: The resulting image after overlay , in PIL format.
    s
    Example:
        image = Image.open(' original .jpg ')
        depth_map = depth(image)
        overlayed_image = overlay_images(depth_map, image, alpha=0.3)
        display (overlayed_image)
    """
```
# GOAL #: Based on the above tools , I want you to reason about how to solve the # USER REQUEST # and generate the actions step by step (each action is a python jupyter notebook code block) to solve the request .
You may need to use the tools above to process the images and make decisions based on the visual outputs of the previous code blocks.
Your visual ability is not perfect , so you should use these tools to assist you in reasoning about the images.

````
````

```
# [six in−context examples here]

# USER REQUEST #:
```


```
Is the red balloon above or below the white balloon? Options: (A) The red balloon is below the white balloon. (B) The red balloon is above the white balloon.

# USER IMAGE stored in image_1 as PIL image.
Now please generate only THOUGHT 0 and ACTION 0 in RESULT. If no action needed, also reply with ANSWER: <your answer> and ends with TERMINATE in the RESULT:

# RESULT #:
```

Thought + Action 0

````
THOUGHT 0: To determine the relative positions of the red and white balloons, I need to detect the balloons in the image. I will use the detection tool to find the balloons.
ACTION 0:
```python
image_1_balloon_detection, image_1_balloon_boxes = detection(image_1, ["red balloon", "white balloon"])
display(image_1_balloon_detection.annotated_image)
```
````

Observation 0

Execution success. The output is as follows.

Please generate the next THOUGHT and ACTION. If you can get the answer, please also reply with ANSWER: <your answer> and ends with TERMINATE.

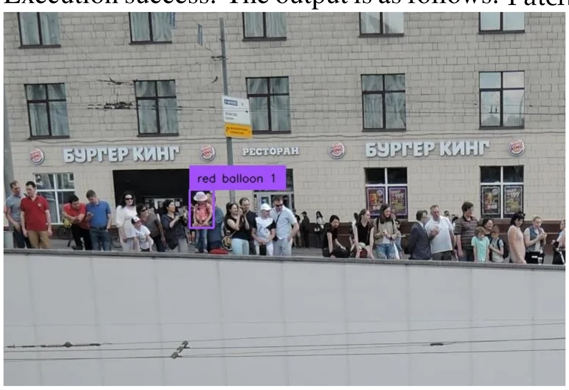
Thought + Action 1

```
THOUGHT 1: The detection tool did not annotate the balloons in the image. I will use the sliding_window_detection tool to search for the red and white balloons in the image.

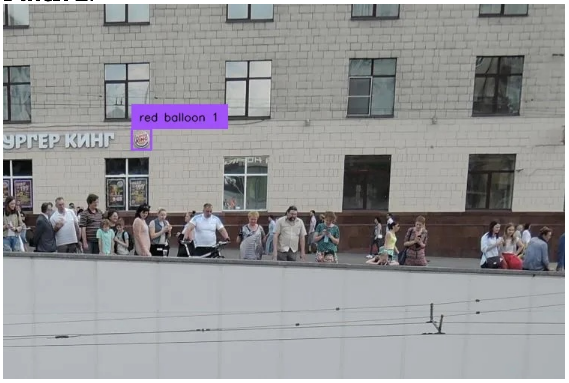
ACTION 1:
```python
possible_patches, possible_boxes = sliding_window_detection(image_1, ["red balloon", "white balloon"])
for i, patch in enumerate(possible_patches):
    print(f"Patch {i+1}:")
    display(patch.annotated_image)
```
```

Observation 1

Execution success. The output is as follows. Patch 1:



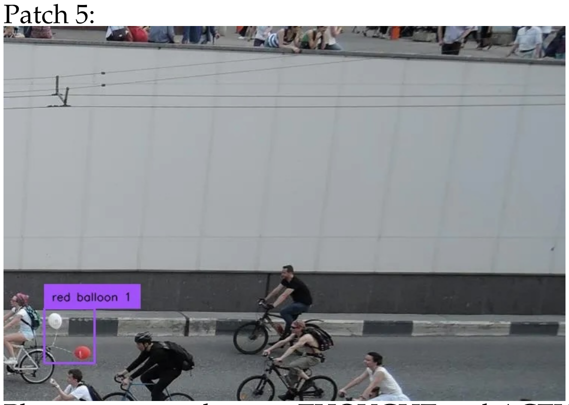
Patch 2:



Patch 3:

Patch 4:

Patch 5:



Please generate the next THOUGHT and ACTION. If you can get the answer, please also reply with ANSWER: <your answer> and ends with TERMINATE.

Thought + Action 2

THOUGHT 2: From Patch 4 and 5, I can see both the red and white balloons. The red balloon is clearly below the white balloon in these patches.
ACTION 2: No action needed.
ANSWER: The red balloon is below the white balloon. So the answer is (A). TERMINATE

Prompts for math tasks. The prompts are similar to the vision task for math tasks, except we remove the computer vision specialists, and add example codes for math plotting. Besides, the user query is different. For each task, the user query part of the prompt is as follows.

PROMPT
You are given a real-valued, scalar function $f(x)$.
YOUR TASK is to determine whether $f(x)$ is an even function, an odd function, or neither. Definition of an odd function: A function such that

$$f(-x) = -f(x)$$

where the sign is reversed but the absolute value remains the same if the sign of the independent variable is reversed.
A function is neither even nor odd if it does not satisfy either condition.
Here is the expression of $f(x)$:

$$f(x) = \frac{-2x^5}{2x^8 - 4x^6 + 12x^4 + 4x^2 + 11.16}$$

Respond with 'even', 'odd', 'neither' first on whether the function $f(x)$ is even, odd, or neither, based on the definitions and your observation of the function. You can generate matplotlib code to visualize the function.
If you can get the answer, please reply with ANSWER: <your answer>, extract the final answer in FINAL ANSWER: <final answer> and ends with TERMINATE in the RESULT.
Answer:

Figure A.1: Prompt for the Math Parity task. We follow a similar prompt format to Fu et al., except prompting the models to write the code to generate images.

PROMPT
You are given a real-valued, scalar function $f(x)$.
YOUR TASK is to determine whether $f(x)$ is an convex function or concave function. Definition of a convex function: A function such that

$$\forall x, y, 0 \leq t \leq 1, f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

Definition of a concave function: A function such that

$$\forall x, y, 0 \leq t \leq 1, f(tx + (1-t)y) \geq tf(x) + (1-t)f(y)$$

Here is the expression of $f(x)$:

$$f(x) = 7.57 - 0.08 * Abs(x)$$

Respond with 'convex' or 'concave' first on whether the function f (x) is convex or concave, based on the definitions and your observation of the function. You can generate matplotlib code to visualize the function.
If you can get the answer, please reply with ANSWER: <your answer>, extract the final answer in FINAL ANSWER: <final answer> and ends with TERMINATE in the RESULT.
Answer:

Figure A.2: Prompt for the Math Convexity task. We follow the similar prompt format to Fu et al., except prompting the models to write the code to generate images.

PROMPT
You are given an adjacency matrix of a graph and two query nodes.
YOUR TASK is to find if there is a path between the two nodes.
Definition of connectivity: In an undirected graph G, two vertices u and v are called connected if G contains a path from u to v. A path in a graph is a finite sequence of edges which joins a sequence of vertices. In the query example, the nodes and the adjacency matrix are zero-indexed.
Query Example:
Adjacency Matrix:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Query nodes indices (zero-indexed): 4 and 0
Respond with 'yes' or 'no' first on whether the query nodes are connected or not in the graph.
If there is a path, first provide the path as a sequence of vertices (nodes), and then explain your reasoning. You can use networkx to draw the graph. If there is no path, explain why in details. Answer (start with 'yes' or 'no'): If you can get the answer, please reply with ANSWER: <your answer>, extract the final answer in FINAL ANSWER: <final answer> and ends with TERMINATE in the RESULT.
Answer:

Figure A.3: Prompt for the Graph Connectivity task. We follow the similar prompt format to Fu et al., except prompting the models to write the code to generate images.

PROMPT
You are given a visual representation of two graphs, graph G on the left and graph H on the right.
YOUR TASK is to determine whether the two graphs are isomorphic to each other.
Definition of graph isomorphism: In graph theory, an isomorphism of graphs G and H is a bijection f between the vertex sets of G and H, denoted as $f : V(G) \to V(H)$. G and H are said to be isomorphic when f satisfies the following: any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. This kind of bijection is commonly described as "edge-preserving bijection", in accordance with the general notion of isomorphism being a structure-preserving bijection.
In the query example, the adjacency matrices are zero-indexed.
Adjacency Matrix G:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Adjacency Matrix H:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Respond with 'yes' or 'no' first on whether the two graphs are isomorphic to each other. You can use networkx to draw the graph. If they are isomorphic, first provide the bijection between the two graphs, and then explain your reasoning. You can use networkx to draw the graph. If they are not isomorphic, explain why in detail. Answer (start with 'yes' or 'no'): If you can get the answer, please reply with ANSWER: <your answer>, extract the final answer in FINAL ANSWER: <final answer> and ends with TERMINATE in the RESULT.
Answer:

Figure A.4: Prompt for the Graph Isomorphism task. We follow a similar prompt format to Fu et al., except prompting the models to write the code to generate images.

PROMPT
You are given an adjacency matrix of a graph and two query nodes (one source node and one sink node). The source node is the node where the flow starts and the sink node is the node where the flow ends.
YOUR TASK is to solve the maxflow problem given the weighted directed graph.
Definition of Maxflow problem: In the max flow problem, we have a directed graph with a source node s and a sink node t, and each edge has a capacity (integer valued, colored in green) that represents the maximum amount of flow that can be sent through it. The goal is to find the maximum amount of flow that can be sent from s to t, while respecting the capacity constraints on the edges.
Query Example:
Adjacency Matrix:

$$\begin{bmatrix} 0 & 1 & 4 \\ 0 & 0 & 6 \\ 0 & 0 & 0 \end{bmatrix}$$

Source node (zero-indexed): 0
Sink node (zero-indexed): 2
In the query example, the nodes and the adjacency matrix are zero-indexed. You can use networkx to draw the graph.
If you can get the answer, please reply with ANSWER: <your answer>, extract the final answer in FINAL ANSWER: <final answer> and ends with TERMINATE in the RESULT.
Answer:

Figure A.5: Prompt for Graph Maxflow task. We follow the similar prompt format to Fu et al., except prompting the models to solve the maxflow problem.

PROMPT
Given the following FEN of the chess game:
1r1q1rk1/1b2b1Qp/4pp1B/pp1nP3/2pPN3/P1P5/1PB3PP/R4RK1 b - - 0 18
Determine the game's outcome. Who won: White or Black? Answer (start with 'white' or 'black' or 'draw'):
You can draw the chess board using Python given the FEN string. If you can get the answer, please reply with ANSWER: <your answer>, extract the final answer in FINAL ANSWER: <final answer> and ends with TERMINATE in the RESULT.
Answer:

Figure A.6: Prompt for Winner ID task. We follow a similar prompt format to Fu et al., except prompting the models to analyze the game outcome.

A.2 Dataset Statistics

Table A.1 and A.2 show the statistics of the datasets we used, including IsoBench Fu et al., BLINK Fu et al. [2024], MMVP Tong et al. [2024], and V^*Bench Wu and Xie [2024].

Dataset	size	partition	representation
Math Parity	383	val	code
Math Convexity	255	val	code
Graph Maxflow	128	val	array
Graph Connectivity	128	val	array
Graph Isomorphism	128	val	array
Winner ID	257	val	FEN

Table A.1: IsoBench Fu et al. data statistics.

Dataset	size	partition	input
$V*$Bench	257	-	Single Image
MMVP	300	-	Single Image
BLINK Relative Depth	124	val	Single Image
BLINK Spatial Relation	143	val	Single Image
BLINK Jigsaw Puzzle	150	val	Multiple Images
BLINK Visual Correspondence	172	val	Multiple Images
BLINK Semantic Correspondence	139	val	Multiple Image

Table A.2: Vision tasks data statistics.

A.3 Costs

The cost of running each task using GPT-4o is in Table A.3.

Dataset	tokens per sample	GPT-4o cost per sample
Math Parity	2994	\$0.015
Math Convexity	2211	\$0.011
Graph Connectivity	2819	\$0.014
Graph Isomorphism	3143	\$0.016
V^*Bench	26647	\$0.133
MMVP	11870	\$0.059
BLINK Relative Detph	14078	\$0.070
BLINK Spatial Relation	12848	\$0.064
BLINK Jigsaw Puzzle	13206	\$0.066
BLINK Visual Correspondence	16988	\$0.085
BLINK Semantic Correspondence	11508	\$0.058

Table A.3: The cost of running Sketchpad on each task.

Chapter B

Appendix for TIFA: Accurate and Interpretable Text-to-Image Faithfulness Evaluation with Question Answering

B.1 Annotation Details

B.1.1 Likert Scale on Text-to-Image Faithfulness

Guidelines The annotation guideline is as follows:

- On a scale of 1-5, score "does the image match the prompt?".
- The ranking of each image given the same text input is important. If you believe the current scoring criteria cannot reflect your ranking preference, pick scores that are consistent with your ranking. Ties are allowed.
- To evaluate the generated image, there are two aspects: image quality and text-image match. Here we only care about text-image match, which is referred to as "faithfulness".
- There are several kinds of elements in the text: object, attribute, relation, and context. Measure the consistency by counting how many elements are missed/misrepresented in the generated image.

- For some elements, e.g. “train conductor’s hat", if you can see there is a hat but not a train conductor’s hat, consider half of the element is missed/misrepresented in the generated image.
- Objects are the most important elements. If an object is missing, then consider all related attributes, activity, and attributes missing.
- When you cannot tell what the object/attribute/activity/context is, consider the element missing. (e.g., can’t tell if an object is a microwave)

Given the above guideline, suppose the text input contains n elements, and x elements are missed or misrepresented. n and x are all counted by the annotators. The reference scoring guideline is as follows:

- 5: The image perfectly matches the prompt.
- 4: $x \leq 2$ and $x \leq n/3$. A few elements are missed/misrepresented.
- 3: $\min\{2, n/3\} < x \leq n/2$ elements are missed/misrepresented.
- 2: $x > n/2$. More than half of the elements are missed/misrepresented.
- 1: None of the major objects are correctly presented in the image.

Details We collect 1600 annotations on 800 generated images from 160 text inputs. Each image is scored by 2 annotators, and we collect the scores from 20 graduate students. We average the scores as the final faithfulness score of the image. The inter-annotator agreement measured by Krippendorf’s α is 0.67, indicating “substantial" agreement. The images are generated by the five most recent text-to-image models in our study, including VQ-Diffusion Gu et al. [2021], minDALL-E Kim et al. [2021a], and Stable Diffusion Saharia et al. [2022] v1.1, v1.5, and v2.1. For each text input, we present the five images together, making it easier for the annotators to give faithfulness scores that reflect their ranking preference. We will release the annotation scores on publication.

B.1.2 Human VQA

Guidelines Given an image, a question, and a set of choices, choose the correct choice according to the image content. There are two types of questions. One has two options: "(A) yes (B) no". Another type of question has four choices. We also add the fifth option “None of the above". If you believe none of the four choices is correct, choose the fifth one. Some images are of low quality. Just select the choice according to your instinct. For ambiguous cases, for example, the question is “is there a man?", and the image contains a human but it is unclear whether the human is a man, answer “no".

Details We collect annotations of 1029 questions on 126 generated images. The images are from images used in the Likert Scale annotation. Each question is answered by two annotators, and we have the same 20 graduate students as the annotators. The inter-annotator agreement measured by Krippendorf's α is 0.88. A third annotator is involved if the two annotators disagree. And the final answer is given by the majority vote. We will release the annotated VQA answers.

Chapter C

Appendix for NavigationBench

C.1 Simulated user system prompt.

The system prompt is as following. The user_requirement, user_preference, context, and time are from the task scenario.

PROMPT
You are simulating a user that is driving on a highway. Now you are chatting with the smart car assistant to add another stop on the map. You have some requirements and preferences for the stop, and for time and distance constraints. For time and distance constraints, you may want the stop to be within a certain time or distance from the current location; Or a stop that adds a certain time or distance detour to the trip; Or a stop that is around a certain time or distance from the current location.

The requirement you have is: {user_requirement}, all these must be satisfied.
Your preference is: {user_preference}. For these, the more the satisfied, the better.
And here is more context: {context}
The current time is {time}.

Rules:
- Don't reveal all preferences at once
- be realistic like a real person
- Answer the assistant's questions.
- Occasionally add new details that seem natural in context
- If the assistant shows you some options, react based on your preferences
- If the assistant says it cannot see information on some of your preferences, skip that preference, like real users.
- Be conversational and natural
- If an option matches your preferences well, show enthusiasm
- You are driving a car, so you can't see the map. You can only hear the assistant's message. Also, you can't say too much because you are driving. Be succint and natural.
- As a user that is driving, you won't say too many words. Be realistic and ends the dialogue in 10 turns. It is ok to relax some preferences.

Recent conversation context will be provided. Respond as this user would.

Figure C.1: System prompt for the simulated user.