Breaking the Language Model Monolith: Towards Augmented and Modular Language Models

Weijia Shi

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington
2026

Reading Committee:
Luke Zettlemoyer, Co-Chair
Noah A. Smith, Co-Chair
Colin Raffel

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Breaking the Language Model Monolith: Towards Augmented and Modular Language Models

Weijia Shi

Co-chairs of the Supervisory Committee:

Professor Luke Zettlemoyer
Computer Science and Engineering

Professor Noah A. Smith
Computer Science and Engineering

Current language models (LMs) are monolithic: very large models are trained on all available data and deployed for every use case. While this paradigm has produced remarkably capable systems, it presents fundamental challenges. Monolithic LMs frequently generate factually incorrect statements because they cannot update knowledge without costly retraining. They memorize sensitive training data, creating serious privacy and copyright risks. And they offer little mechanism for data owners to control whether or how their data is used in model training and inference. This dissertation introduces modularity as a principled solution to the limitations of the monolithic paradigm. Rather than treating a language model as a single blackbox, we argue for architectures and training algorithms that separate capabilities across composable components. My research spans three areas while adding modularity across the entire LM stack, from pretraining and post-training to analysis: (1) **Understanding the limits of monolithic LMs.** We investigate the limits of monolithic LMs by studying how training data influences model behavior. (2) **Model modularity.** We develop methods for model modularity, augmenting language models with external modules to improve factuality, breadth of knowledge, and reasoning. (3) **Data modularity.** We design new ways to co-develop LMs that are distributed and data modular. Across these areas, modularity offers a path toward language models that are more factual, more transparent about their training data, and easier to adapt after training.

Acknowledgements

There are many people without whom my PhD journey would not have been possible. First and foremost, I would like to thank my advisors, Luke Zettlemoyer and Noah Smith. I still remember how excited I was when I received my offer from the University of Washington. During my undergraduate years, one of my main projects involved retrofitting ELMo embeddings, and I vividly remember how mind-blowing it felt to read the ELMo paper for the first time. So when I got Luke's email, I accepted very quickly.

During the first two years of my PhD, I was not entirely sure what I wanted to work on and spent a lot of time exploring different directions. Luke was always patient, giving me an extraordinary amount of freedom to explore while guiding me toward meaningful and impactful problems. His trust gave me the confidence to develop my own research taste and independence. Later, I had the opportunity to work closely with Noah, who brought a different and invaluable kind of mentorship. One of Noah's superpowers is his ability to grasp new ideas with extraordinary speed. Whenever I shared a research update with Noah, he could always quickly identify the core of an idea, transform it into a sharp research question, and guide me on how to build it into a strong project. Early in my PhD, I would often go into advisor meetings feeling like I needed to prove I had worked hard enough that week. Over time, I came to understand that this was never what they cared about. What mattered to both Luke and Noah was helping me grow into an independent researcher and supporting me through everything that entailed.

I would also like to thank my committee members, Colin Raffel and Lucy Li. Colin is a researcher I have long admired. His research, talks, and blog posts have had a profound influence on me throughout my PhD. When I felt uncertain about my career, I have returned to his writing many times, which has been a real source of encouragement for me.

Before my PhD, I was fortunate to have mentors during my undergraduate years, Kai-Wei Chang, Dan

Roth, and Adnan Darwiche, who taught me how to do research when I had no experience. I am deeply grateful for the foundation they gave me.

I would also like to thank Pang-Wei Koh, Ranjay Krishna, and Danqi Chen. Watching them build thriving research groups from the ground up has been inspiring. They have shown me what it looks like to pursue ambitious ideas, mentor students thoughtfully, and create environments where people can do their best work. I am also grateful to Yulia Tsvetkov and Hannaneh Hajishirzi for their advice and encouragement throughout my time at UW.

Toward the end of my second year, I began working part-time at Meta, an experience that turned out to be life-changing. There, I had the opportunity to work alongside some of the most brilliant researchers I have ever encountered: Mike Lewis, Scott Yih, Chunting Zhou, and Lili Yu. In many ways, they became third advisors to me. Whenever I had a new research idea, they were the first people I would seek out. I learned enormously from how they approached problems, asked questions, and guided projects. Watching Chunting and Lili navigate the many challenges behind the Transfusion project and push out such impactful work taught me a lot. Even now, though I no longer work there, I still visit the office from time to time, which reminds me how much that experience shaped me, both as a researcher and as a person.

During my PhD, I had the pleasure of working closely with Xiaochuang Han and Yushi Hu, both of whom are extraordinarily insightful. I still remember the late nights we spent working before conference deadlines at the university or at the Meta office. Those moments were often stressful, but they have become some of my most cherished memories from graduate school. I would also like to thank my collaborators Shangbin Feng, Sewon Min, Akari Asai, Daogao Liu, Yangsibo Huang, Chiyuan Zhang, Hongjin Su, Mengzhou Xia, Niklas Muennighoff, Michihiro Yasunaga, Zeqiu Wu, Fangyuan Xu, Boyi Wei, Lucy He, Victor Zhong, Yizhong Wang, Tao Yu, Tim Dettmers, Suchin Gururangan and Eunsol Choi. I deeply admire their work ethic, research vision, and commitment to research.

The PhD journey would not have been nearly as enjoyable without the friends I made at UW: Thao Nguyen, Inna Lin, Melanie Sclar, Orevaoghene Ahia, Alisa Liu, Jon Hayase, Jiacheng Liu, Ari Holtzman, Margaret Li, Jaehun Jung, and Yunbi Kim and Artidoro Pagnoni. Many of us started our PhDs around the same time and rode the ups and downs of graduate school together. We can talk through everything from research, life and random worries. Sharing this journey with all of you made the hard moments easier and the

good ones more memorable.

I would also like to give special thanks to Preston Jiang, Rock Pang, Tuochao Chen, and Jieyu Zhang. Preston, Rock, and Tuochao have heard me talk about "breaking the monolith" so many times that they could probably give the talk on my behalf. Their warmth and humor brought so much joy to my day-to-day life. Jieyu and I started our PhDs in the same year, but he has always seemed one step ahead. During our junior years, he worked like a senior PhD student; during our senior years, he worked like a professor. I learned a lot from his maturity, thoughtfulness, and approach to research.

During the second half of my PhD, I became friends with a new generation of UW NLP students: Stella Li, Rulin Shao, Zixian Ma, Hamish Ivison, Jacqueline He, Oscar Yinn, Tong Chen, Rui Xin, Scott Geng. What struck me most was how hard all of them worked while also really knowing how to have fun. I will miss the late-night gatherings and concerts we shared.

I would also like to thank friends outside of UW who supported me throughout this journey: Xingyu Fu, Shirley Wu, Lisa Li, Alex Gu, and Naman Jain. We met at different stages, but all became my lifelong friends. More broadly, I am grateful to the many members of ZLab, Noah's Ark Lab, and the wider UW NLP community. The collaborative and supportive environment they collectively created made UW an exceptional place to grow as a researcher.

Finally, I would like to thank my parents. Throughout my life they have supported me unconditionally and encouraged me at every turn.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Current language models (LMs) are monolithic: very large models are trained indiscriminately on all available data and subsequently deployed for every use case [Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023; Dubey et al., 2024; Team, 2023]. The scale of these systems is remarkable: modern LMs are trained on trillions of tokens and contain trillions of parameters, encoding an enormous breadth of knowledge entirely within their parameters. This paradigm has produced language models of unprecedented capability: they can answer complex questions, write code, translate between languages, and reason over multimodal inputs such as images and video. However, the monolithic design carries deep structural limitations. When all knowledge is fused into a single set of parameters, the resulting systems face three fundamental challenges. (1) *Factual errors and staleness.* Monolithic LMs encode a static snapshot of knowledge at training time. As the world changes, the model's knowledge becomes stale, but updating it requires expensive retraining. And because knowledge is spread diffusely across billions of parameters, these models frequently generate factually incorrect statements [Ji et al., 2023]. (2) *Privacy and copyright issues.* When trained on massive web-scraped corpora, monolithic LMs inevitably memorize sensitive data: personal emails, medical records, copyrighted books, and private conversations [Carlini et al., 2021]. Once memorized, such information is extremely difficult to remove without retraining from scratch. This creates serious risks for individuals whose data appears in training corpora and for organizations subject to privacy regulations. Copyright holders face similar challenges, as models may generate verbatim excerpts from copyrighted texts they have memorized. (3) *Opacity and lack of control.* The decisions made by monolithic LMs are opaque because it is difficult

to trace which training data influenced any particular output. Data owners have no practical mechanism to verify whether their data was used in training, to audit its influence, or to request its removal. This opacity makes responsible deployment and regulatory compliance extremely challenging.

My research focuses on *modularity* as a principled solution to these limitations, where capabilities are separated into independently developed, reusable components. Rather than treating a language model as a single, static blackbox, modular systems separate capabilities across composable components that can be independently developed, inspected, updated, and composed at inference time. Modularity manifests at two levels:

- **Model modularity**: augmenting existing LMs with external modules, retrievers, compressors, and specialist vision models, that dynamically supply knowledge and capabilities at inference time.
- **Data modularity**: designing the LM itself from decentralized components, trained on different datasets, that can be co-developed and flexibly composed while preserving data privacy.

This thesis is structured as follows: Before introducing modularity as a solution, we first characterize the limitations of the monolithic paradigm (Chapter 2), focusing on how training data is memorized and what the consequences are for privacy and copyright. A central question in responsible LM deployment is whether models have memorized specific training data, an issue with profound implications for privacy, copyright, and benchmark validity. The datasets used to train today's largest LMs are rarely disclosed, making systematic auditing extremely difficult. To address this, we adapt membership inference attacks [Shokri et al., 2017] from the computer privacy literature to study pretraining data detection in large LMs. Beyond detecting memorization, a natural response is to attempt to *remove* it through techniques like machine unlearning. Our systematic study shows that current methods still fail to meet basic expectations around privacy, utility, and scalability. These findings highlight that practical unlearning for LMs remains an open problem, motivating our work on modular architectures that can address data control at a more fundamental architectural level.

Model modularity (Chapter 3) develops methods for augmenting language models with external modules, retrievers and specialist vision models, to improve factuality and reasoning. Rather than storing all knowledge in model parameters, we maintain knowledge in an external datastore queried at inference time. We develop three main contributions: a framework for black-box LMs with plug-and-play retrieval, together with a

training scheme that adapts the retriever to maximize the LM's benefit from retrieved content; a pretraining method that trains LMs on topically related document sequences rather than random concatenations, significantly improving performance on knowledge-intensive tasks; and a training-free decoding method that encourages faithful attention to retrieved context. We additionally describe two further contributions more briefly: an instruction-finetuned embedding model that generates task-tailored embeddings from explicit task descriptions, and a document-compression method that condenses retrieved passages to query-relevant content and selectively skips retrieval when unnecessary.

We further extend modularity to the visual domain (Chapter 4), augmenting LMs with specialist vision models as reasoning tools. Our main contribution equips multimodal LMs with the ability to produce visual artifacts such as supporting lines and sketches as intermediate reasoning steps, enabling specialist vision tools to serve as a visual chain-of-thought for spatial and perceptual tasks. We also briefly cover retrieval-augmented multimodal generation, which grounds image generation in retrieved image-text references rather than relying on parametric knowledge alone.

Finally, data modularity (Chapter 5) challenges the assumption that all parameters must be trained on centralized data. We introduce a mixture-of-experts training framework that enables distributed training on locally maintained private datasets: each data owner independently trains an expert, and these experts can be merged without any joint training or data sharing. We further propose an efficient multimodal architecture that composes a pretrained text LM with dedicated image-processing modules via modality separation, preserving existing language capabilities while adding visual understanding at a fraction of the compute cost of training from scratch.

Chapter 2

Understanding the Limits of Monolithic Language Models

Training modern LMs requires corpora on the order of trillions of tokens [Brown et al., 2020; Dubey et al., 2024], assembled primarily by crawling the open web at scale. At this scale, such corpora inevitably absorb sensitive personal information, copyrighted text, and held-out evaluation benchmarks. Because all of this data is used to update a single set of parameters, the model may implicitly encode specific training sequences and, when prompted, reproduce them verbatim [Carlini et al., 2021], a phenomenon known as *memorization*.

This raises two practical questions. First, given that training data is rarely disclosed, how can affected parties, individuals, rights-holders, or evaluators, determine whether their data was used to train a model? Second, once memorization is identified, can it be reliably removed without retraining from scratch? These questions have implications for privacy regulation, copyright enforcement, and the integrity of model evaluation. Before proposing solutions, we need to understand the depth of the problem. Understanding *how* memorization can be detected and *whether* it can be removed informs the design of architectural alternatives and helps set the right expectations.

This chapter presents two lines of empirical investigation. First, we study the *data detection problem*: given a piece of text and black-box access to an LM, can we determine whether the model was trained on it? Second, we study the *data removal problem*: once memorization is detected, can we reliably remove it? Together, the answers paint a concerning picture: memorization can be detected at scale (§2.1), and current

removal algorithms fall short of basic expectations (§2.2). This motivates the architectural alternatives in the chapters that follow.

2.1 Detecting Pretraining Data from Large Language Models

Weijia Shi*, Anirudh Ajith*, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting Pretraining Data from Large Language Models. In *International Conference on Learning Representations*.

2.1.1 Motivation

As LM training corpora have grown to encompass trillions of tokens, model developers have become increasingly reluctant to disclose the full composition or sources of their training data. This opacity creates critical challenges. First, personally identifiable information and copyrighted materials are almost certainly present in any sufficiently large web-scraped corpus, and there is currently no way for individuals or rights-holders to verify whether their data was used. Second, benchmark contamination, when evaluation test data appears in the training corpus, makes it difficult to accurately assess model capabilities [Magar and Schwartz, 2022].

We study the *pretraining data detection problem*: given a piece of text and black-box access to an LM, can we determine whether the model was trained on that text? This is an instance of membership inference attack (MIA) [Shokri et al., 2017] applied to the pretraining setting.

2.1.2 The WikiMIA Benchmark

Adapting existing MIA methods to pretraining data detection faces a fundamental benchmarking challenge: obtaining ground-truth labels (member vs. non-member) for pretraining data without access to the actual training set. Most prior MIA benchmarks assume access to the training distribution or shadow models, which is impractical for large frontier LMs.

We introduce **WIKIMIA**, a dynamic benchmark that exploits the temporal nature of Wikipedia event articles to construct reliable ground-truth labels without accessing the training corpus. The key insight is

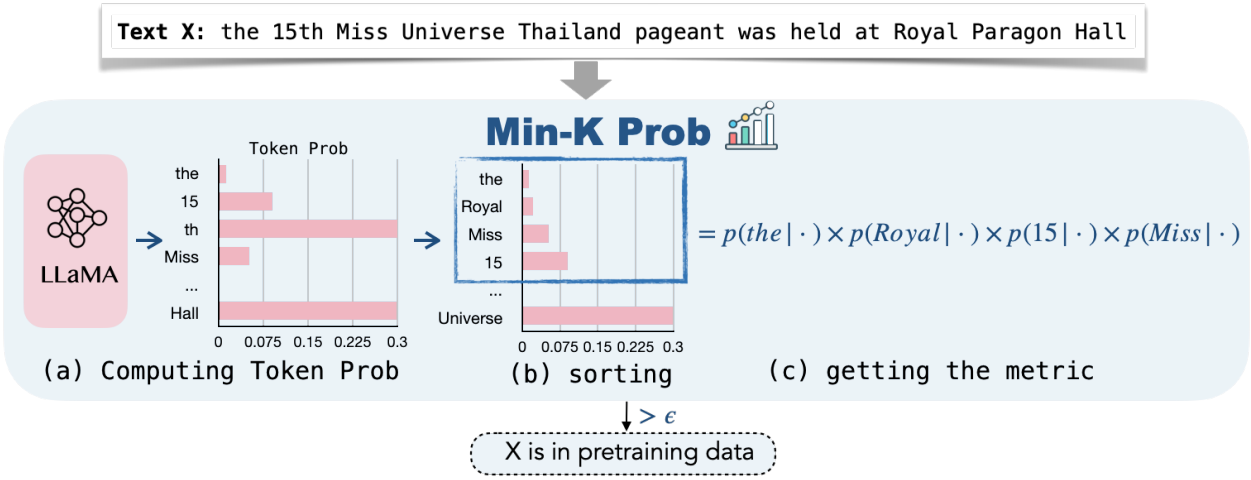

Figure 2.1: Overview of pretraining data detection with MIN-K% PROB. Given a text sequence and black-box LM access, MIN-K% PROB computes the average log-probability of the $k\%$ lowest-probability tokens. Non-member texts tend to contain a few outlier low-probability tokens; member texts do not. This reference-free signal achieves strong detection performance on the WIKIMIA benchmark across diverse LMs.

simple: events that occurred *after* an LM's training cutoff date cannot possibly appear in its pretraining data.

We therefore:

1. Select Wikipedia event articles describing events *before* the LM's training cutoff as **member** data (guaranteed to be in the training corpus, as Wikipedia is a standard pretraining source).

2. Select Wikipedia event articles describing events *after* the cutoff as **non-member** data (guaranteed not to be in the training corpus).

This construction ensures three desirable properties:

- **Accuracy**: non-member labels are guaranteed by the temporal constraint, events after the cutoff cannot appear in training data.

- **Generality**: Wikipedia is a standard component of nearly all LM pretraining corpora (OPT, LLaMA, GPT-Neo, etc.), making the benchmark applicable across many models.

- **Dynamism**: new non-member data is continuously generated as Wikipedia covers new events, allowing the benchmark to evaluate newly released models without requiring label collection.

2.1.3 The Min-k% Prob Method

Existing MIA methods for fine-tuning data detection rely on reference models trained on data similar to the target model's training distribution [Carlini et al., 2022]. This is impractical in the pretraining setting: the pretraining distribution is unknown, and training a reference model at the scale of frontier LMs is computationally prohibitive. We propose **MIN-K% PROB**, a reference-free detection method. Our method is grounded in the following observation:

> *An unseen (non-member) example tends to contain a few tokens that the model assigns unusually low probability to, outlier tokens that the model has never seen in this particular context. A seen (member) example is less likely to contain such outlier tokens, because the model has learned to predict the full sequence.*

Formally, given a text sequence $X = (x_1, \ldots, x_n)$ and a language model p_θ, MIN-K% PROB computes the average log-probability of the $k\%$ tokens with the lowest probabilities:

$$\text{MIN-K\% PROB}(X) = \frac{1}{|S|} \sum_{x_i \in S} \log p_\theta(x_i \mid x_{<i}), \tag{2.1}$$

where S is the set of the $k\%$ tokens with lowest $\log p_\theta(x_i \mid x_{<i})$.

The intuition is that seen examples are more likely to be recalled reliably by the model, while unseen examples often contain local surprises, unusual word choices, proper nouns, or rare phrasings, that the model assigns low probability to. By focusing on the minimum-probability tokens rather than the average, MIN-K% PROB amplifies this signal.

Advantages over baselines. MIN-K% PROB requires only the model's output probabilities, no shadow models, no reference models, no access to model internals beyond the probability API. This makes it directly applicable to any black-box LM accessible through a standard token probability API. Compared to the strongest prior reference-free baseline (average log-likelihood), MIN-K% PROB achieves a 7.4% improvement in AUC on WIKIMIA.

We also empirically analyze the effect of model size and text length on detection difficulty. Detection performance improves with model size, larger models memorize more aggressively, and with text length,

Table 2.1: AUC scores for detecting pretraining examples on WIKIMIA. *Ori.* and *Para.* are the original and paraphrased settings. **Bold** = best per column. MIN-K% PROB outperforms all reference-free baselines across all five models.

	Pythia-2.8B		NeoX-20B		LLaMA-30B		LLaMA-65B		OPT-66B		
Method	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	**Avg.**
Neighbor	0.61	0.59	0.68	0.58	0.71	0.62	0.71	0.69	0.65	0.62	0.65
PPL	0.61	0.61	0.70	0.70	0.70	0.70	0.71	0.72	0.66	0.64	0.67
Zlib	0.65	0.54	0.72	0.62	0.72	0.64	0.72	0.66	0.67	0.57	0.65
Lowercase	0.59	0.60	0.68	0.67	0.59	0.54	0.63	0.60	0.59	0.58	0.61
Smaller Ref	0.60	0.58	0.68	0.65	0.72	0.64	0.74	0.70	0.67	0.64	0.66
MIN-K% PROB	**0.67**	**0.66**	**0.76**	**0.74**	**0.74**	**0.73**	**0.74**	**0.74**	**0.71**	**0.69**	**0.72**

longer sequences provide more tokens for the outlier signal to emerge from. Additional results across the OPT, GPT-Neo, and LLaMA-1 families, a sensitivity analysis of the k parameter, and a detailed text-length breakdown are reported in Appendix A (Tables A.1–A.3).

2.1.4 Case Studies

Copyrighted book detection. We apply MIN-K% PROB to the Books3 dataset [Gao et al., 2021], a collection of copyrighted books that has been included in some LM pretraining corpora. Applying MIN-K% PROB to GPT-3 (`text-davinci-003`), we find strong evidence that GPT-3 was pretrained on a substantial portion of Books3 content, with significantly higher detection scores for books that are known to be in common pretraining corpora. This is direct evidence that a commercial frontier model memorized copyrighted books.

Benchmark contamination detection. We simulate a benchmark contamination scenario by constructing synthetic pretraining corpora that include examples from six standard downstream evaluation datasets. MIN-K% PROB reliably detects contamination and allows us to quantify how detection difficulty varies with the contamination rate, the frequency of contaminated examples, and the pretraining learning rate. Higher learning rates and more frequent exposures lead to stronger memorization and easier detection.

Privacy auditing of machine unlearning. We apply MIN-K% PROB as an auditing tool to evaluate machine unlearning algorithms that have been applied to remove copyrighted books from a trained LM [Eldan

and Russinovich, 2023]. Even after applying the unlearning algorithm, MIN-K% PROB detects statistical traces of the unlearned content in the model, demonstrating that approximate unlearning does not fully eliminate memorization. This finding directly motivates our subsequent work developing MUSE, a more comprehensive evaluation framework for machine unlearning.

Community impact. The Open LM Leaderboard [Hugging Face, 2023] integrated contamination detection tools based on this work, encouraging model developers to audit their training pipelines before public release. Our work opened a new research direction in LM data auditing that has since been extensively followed up.

2.2 Machine Unlearning Six-Way Evaluation

Weijia Shi*, Jaechan Lee*, Yangsibo Huang*, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2025. MUSE: Machine Unlearning Six-Way Evaluation for Language Models. In *International Conference on Learning Representations*.

2.2.1 Motivation

Detecting that data was memorized is the first step; removing that memorization is the next challenge. Machine unlearning for LMs aims to modify a trained model so that it behaves as though certain training data was never seen. Exact unlearning (retraining from scratch without the forgotten data) is the gold standard but is prohibitively expensive for large LMs, especially if unlearning requests arrive frequently. Efficient approximate unlearning algorithms have been proposed [Eldan and Russinovich, 2023; Zhang et al., 2024; Ilharco et al., 2023], but existing evaluation frameworks are narrow and do not capture the full spectrum of what unlearning should achieve.

Concretely, consider a scenario where the author of the Harry Potter book series requests that their copyrighted books be removed from a deployed LM. What does successful unlearning require? Prior evaluations [Eldan and Russinovich, 2023; Maini et al., 2024] test whether the model can still correctly answer factual questions about Harry Potter. However, the author's primary concern is copyright infringement, which is about verbatim reproduction rather than factual knowledge. A more stringent requirement is that the model should not reveal, through statistical tests, whether the Harry Potter books were ever in its training

data. On the deployer's side, the unlearning algorithm should work reliably across requests of varying size (individual passages, chapters, entire books) and should remain effective as multiple unlearning requests accumulate over time.

We propose **MUSE** (Machine Unlearning Six-Way Evaluation), a systematic framework that captures these diverse requirements across six dimensions.

2.2.2 The Six Desiderata

MUSE evaluates machine unlearning algorithms along the following six dimensions (Figure 2.2):

From the data owner's perspective:

1. **No verbatim memorization.** The unlearned model should not be able to reproduce verbatim excerpts from the forget set. We measure this by prompting the model with the beginning of a passage and checking whether it generates the continuation.

2. **No knowledge memorization.** The model should not retain factual knowledge encoded in the forget set. We measure this via question-answering accuracy on facts from the forgotten content.

3. **No privacy leakage.** The model should not reveal, through statistical tests, whether the forget set was in its training data. We measure this using MIN-K% PROB as a membership inference probe. This criterion cannot be satisfied simply by degrading performance on the forget set: a model that assigns anomalously low probability to forget-set content, due to aggressive unlearning, can itself be detected as having undergone unlearning, which leaks membership information.

From the model deployer's perspective:

4. **Utility preservation.** Unlearning should not degrade the model's general capabilities. We measure this via perplexity on standard corpora and performance on downstream tasks from the retain set.

5. **Scalability.** The unlearning algorithm should remain effective across forget sets of varying sizes, from individual paragraphs to entire books. We measure how performance on the six criteria changes as the forget set size increases.

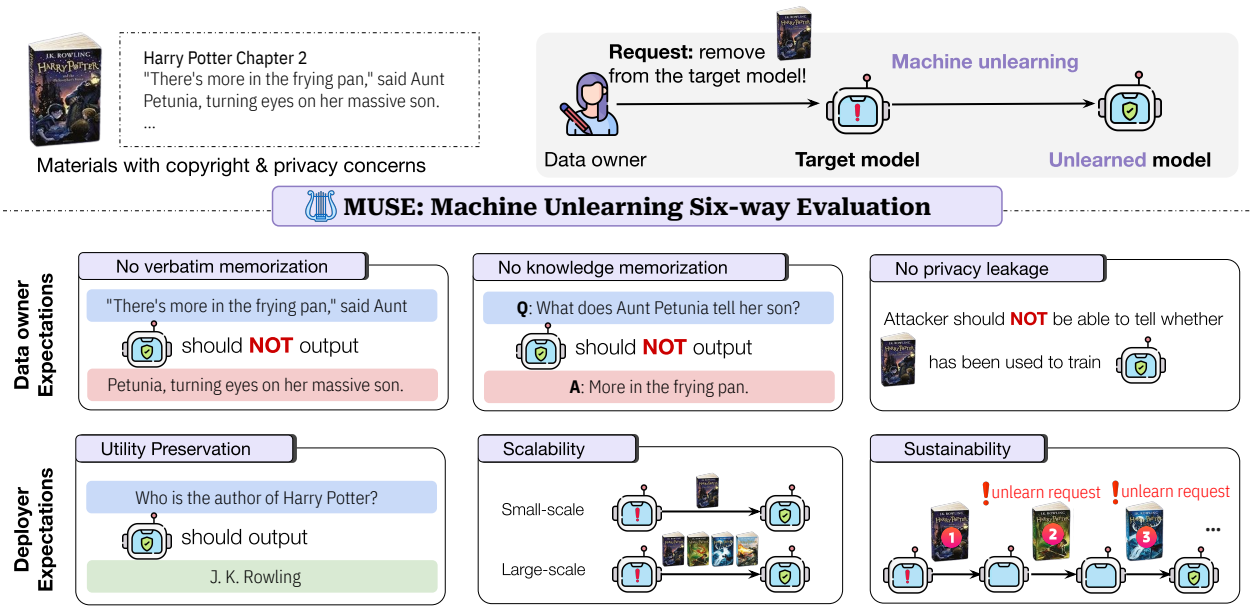

Figure 2.2: MUSE evaluates machine unlearning along six dimensions that jointly capture the expectations of data owners (no verbatim memorization, no knowledge memorization, no privacy leakage) and model deployers (utility preservation, scalability, sustainability).

6. **Sustainability.** The algorithm should remain effective as sequential unlearning requests accumulate. We measure degradation in utility preservation and privacy leakage as multiple batches of content are sequentially unlearned.

2.2.3 Benchmark Construction

We instantiate MUSE for two representative unlearning scenarios:

Books. We use the Harry Potter book series as the forget set, with fan-written Harry Potter content (which shares narrative elements but not verbatim text) as a challenging retain set. This scenario tests the realistic case where the forget and retain sets have significant semantic overlap.

News. We use recent BBC news articles as the forget set, with other BBC articles as the retain set. This scenario tests scalability, as the forget set can range from a few articles to hundreds.

For each scenario, we evaluate eight representative unlearning algorithms spanning gradient ascent, task vectors [Ilharco et al., 2023], negative preference optimization (NPO) [Zhang et al., 2024], and various

regularized variants.

2.2.4 Findings

Our evaluation reveals several critical findings that challenge the state of the art in machine unlearning:

Privacy leakage is nearly universal. Despite successfully reducing verbatim and knowledge memorization, almost all evaluated algorithms fail to prevent statistical privacy leakage. The key insight is that aggressive unlearning causes the model to assign anomalously low probability to forget-set content, lower than a model that had never seen the data. This anomaly is detectable by membership inference, causing the model to *leak* the information that it had previously been trained on this data. Only exact retraining achieves true privacy-preserving unlearning.

Utility and forgetting trade off sharply. The most effective algorithms for forgetting (NPO and task vectors) cause the largest drops in general model utility, making them impractical for deployed systems. Algorithms that preserve utility tend to under-forget, leaving measurable traces of the forgotten content.

Scalability and sustainability are fundamental bottlenecks. As the forget set grows from paragraphs to books, or as sequential unlearning requests accumulate, all algorithms degrade significantly. The utility-forgetting trade-off worsens dramatically at scale. The full six-criterion profile for all eight algorithms, including the scalability and sustainability measurements (C5 and C6) summarized here, is reported in Appendix B (Table B.1); an analysis of how forget/retain topical overlap affects forgetting is given in Table B.2.

Impact. The MUSE benchmark has been adopted by the Open Unlearning initiative as a standard evaluation framework for rigorous, transparent model unlearning research. Our findings highlight that the current suite of unlearning algorithms is not ready for practical deployment in scenarios involving privacy regulations or copyright enforcement, and identify the development of algorithms that can simultaneously achieve all six desiderata as a pressing open problem.

Table 2.2: Most unlearning methods remove memorization but fail on privacy and utility. MUSE evaluates 8 algorithms on C1 (no verbatim mem.), C2 (no knowledge mem.), C3 (no privacy leak.), and C4 (utility preservation) for NEWS and BOOKS corpora. Ratios in blue satisfy the criterion; ratios in orange do not.

	C1. No Verbatim Mem. VerbMem on $\mathcal{D}_u$ (↓)		**C2. No Knowledge Mem.** KnowMem on $\mathcal{D}_u$ (↓)		**C3. No Privacy Leak.** PrivLeak ($\in [-5\%, 5\%]$)		**C4. Utility Preserv.** KnowMem on $\mathcal{D}_r$ (↑)	
NEWS								
Target f_t	58.4		63.9		−99.8		55.2	
Retrain f_r	**20.8**		**33.1**		**0.0**		**55.0**	
GA	0.0	↓100%	0.0	↓100%	5.2	over-unlearn	0.0	↓100%
GA_{GDR}	4.9	↓76.5%	31.0	↓6.3%	108.1	over-unlearn	27.3	↓50.3%
GA_{KLR}	27.4	↑31.4%	50.2	↑51.5%	−96.1	under-unlearn	44.8	↓18.5%
NPO	0.0	↓100%	0.0	↓100%	24.4	over-unlearn	0.0	↓100%
NPO_{GDR}	1.2	↓94.4%	54.6	↑64.8%	105.8	over-unlearn	40.5	↓26.3%
NPO_{KLR}	26.9	↑29.0%	49.0	↑48.1%	−95.8	under-unlearn	45.4	↓17.4%
Task Vector	57.2	↑174.7%	66.2	↑100%	−99.8	under-unlearn	55.8	↑1.5%
WHP	19.7	↓5.6%	21.2	↓35.9%	109.6	under-unlearn	28.3	↓48.5%
BOOKS								
Target f_t	99.8		59.4		−57.5		66.9	
Retrain f_r	**14.3**		**28.9**		**0.0**		**74.5**	
GA	0.0	↓100%	0.0	↓100%	−25.0	under-unlearn	0.0	↓100%
GA_{GDR}	0.0	↓100%	0.0	↓100%	−26.5	under-unlearn	10.7	↓85.6%
GA_{KLR}	16.0	↑11.4%	21.9	↓24.4%	−40.2	under-unlearn	37.2	↓50.0%
NPO	0.0	↓100%	0.0	↓100%	−24.3	under-unlearn	0.0	↓100%
NPO_{GDR}	0.0	↓100%	0.0	↓100%	−30.8	under-unlearn	22.8	↓69.4%
NPO_{KLR}	17.0	↑18.2%	25.0	↓13.4%	−43.5	under-unlearn	44.6	↓40.1%
Task Vector	99.7	↑595%	52.4	↑81.2%	−57.5	under-unlearn	64.7	↓13.1%
WHP	18.0	↑25.2%	55.7	↑92.9%	56.5	over-unlearn	63.6	↓14.6%

2.3 Related Work

Membership inference attacks. The problem of inferring training set membership from model outputs was introduced by Shokri et al. [Shokri et al., 2017] for classification models, using shadow models trained to mimic the target model. Carlini et al. [Carlini et al., 2022] proposed likelihood ratio tests with reference models as a more calibrated alternative. Both approaches assume access to models trained on similar distributions for calibration, which is impractical for large LMs. Our MIN-K% PROB method [Shi et al., 2024a] avoids this requirement entirely, using only the statistical properties of the target model's output probabilities.

Memorization in language models. Carlini et al. [Carlini et al., 2021] demonstrated that large language models can be induced to reproduce verbatim training data through targeted prompting, raising concerns about privacy and copyright. Subsequent work has characterized how memorization depends on data frequency, sequence length, and model scale. Our WIKIMIA benchmark provides a principled methodology for constructing ground-truth member/non-member labels without accessing the training corpus, enabling evaluation of detection methods across models regardless of training data disclosure.

Machine unlearning for LMs. Gradient ascent on the forget set [Jang et al., 2023] is the most direct approach, increasing the model's loss on forgotten content. Eldan and Russinovich [Eldan and Russinovich, 2023] use targeted fine-tuning to replace factual associations. Task vectors [Ilharco et al., 2023] represent fine-tuning as parameter-space vectors and subtract them to "un-learn." Negative Preference Optimization (NPO) [Zhang et al., 2024] reframes unlearning as learning to disprefer the forget set. The TOFU benchmark [Maini et al., 2024] proposed a fictitious-person unlearning task to study forgetting in a controlled setting. MUSE [Shi et al., 2025c] evaluates all these approaches on realistic scenarios (Harry Potter copyright removal, news article removal), revealing their shared failure mode: none can prevent statistical membership inference, even when verbatim and knowledge memorization are successfully removed.

2.4 Conclusion

The two works in this chapter reveal complementary facets of the limits of the monolithic LM paradigm.

MIN-K% PROB shows that memorization of training data is pervasive and detectable, even in black-box models, through simple statistical tests on token probabilities. Applied to GPT-3, it reveals strong evidence of copyrighted book memorization and widespread benchmark contamination. The Open LM Leaderboard has integrated contamination detection tools based on this work, directly improving evaluation integrity for the research community.

MUSE shows that the natural mitigation, removing memorized content via machine unlearning, is far more difficult than the current literature suggests. Of eight evaluated algorithms, none can simultaneously satisfy all six desiderata that a practical unlearning system must meet. The critical gap is statistical privacy leakage: aggressive forgetting causes anomalously low probabilities on the forget set, which is itself a

detectable signal of prior membership. Only retraining from scratch achieves true unlearning, which is computationally prohibitive for large LMs.

Together, these findings motivate a different architectural philosophy: rather than training monolithic models that memorize everything and then attempting surgical removal, we should design systems where data control is built in from the start. This philosophy drives our work on data modularity in Chapter 5, where FLEXOLMO enables independent, privacy-preserving training of expert modules that can be cleanly added or removed at inference time without any surgical unlearning.

2.5 Limitations

WikiMIA benchmark scope. WIKIMIA constructs reliable ground-truth labels by exploiting Wikipedia's temporal structure, but this limits the benchmark to Wikipedia-style text. It may not capture memorization patterns for other data types that are commonly included in pretraining corpora but not well-represented on Wikipedia, such as code, dialogue, or low-resource language text.

Detection of paraphrased content. MIN-K% PROB detects verbatim or near-verbatim memorization through token probability statistics, but is less effective at detecting memorization of paraphrased, summarized, or translated content. In practice, copyright infringement may occur through such transformed reproductions, which the current method would not reliably detect.

MUSE benchmark scenarios. The MUSE benchmark focuses on two representative unlearning scenarios (Harry Potter books and BBC news). While these scenarios cover important real-world cases, they may not fully represent all practical unlearning scenarios (e.g., individual user data deletion as mandated by GDPR, code copyright removal, or medical record privacy). Developing scenario-specific unlearning benchmarks remains an important direction.

MUSE's evaluation of statistical leakage. MUSE's privacy leakage criterion measures whether a membership inference probe (MIN-K% PROB) can detect the forget set in the unlearned model. This is a conservative criterion: an algorithm that achieves low MIN-K% PROB scores on the forget set may still leak membership

through other statistical tests not captured by MIN-K% PROB. A comprehensive privacy evaluation would require a battery of membership inference probes.

Chapter 3

Model Modularity: Augmented Language Models

Chapter 2 established that monolithic LMs memorize training data in ways that are detectable and difficult to remove. This chapter develops the first solution: *model modularity through retrieval augmentation*, which separates the storage of knowledge (in an updatable external datastore) from the LM's core reasoning capabilities (in its parameters). When knowledge is stored externally rather than baked into parameters, it can be updated without retraining, traced back to explicit sources, and selectively restricted or removed.

Building a complete retrieval-augmented system requires solving four interlocking problems: (1) retrieving the right documents in the first place, (2) training LMs that can effectively use retrieved documents, (3) managing the cost and noise of long retrieved passages, and (4) ensuring the LM actually attends to retrieved evidence during inference. This chapter develops three contributions in depth, each targeting one of these problems: REPLUG (§3.1), which establishes the framework for applying retrieval augmentation to black-box LMs and tackles (1); In-Context Pretraining (§3.2), a pretraining algorithm for (2); and Context-Aware Decoding (§3.3), a decoding method for (4). We additionally contribute two methods that we describe only briefly here: INSTRUCTOR [Su et al., 2023], an instruction-finetuned embedding model that produces task-aware embeddings to improve retrieval quality (1), and RECOMP [Xu et al., 2024], which compresses retrieved passages and learns to selectively skip retrieval to manage cost and noise (3).

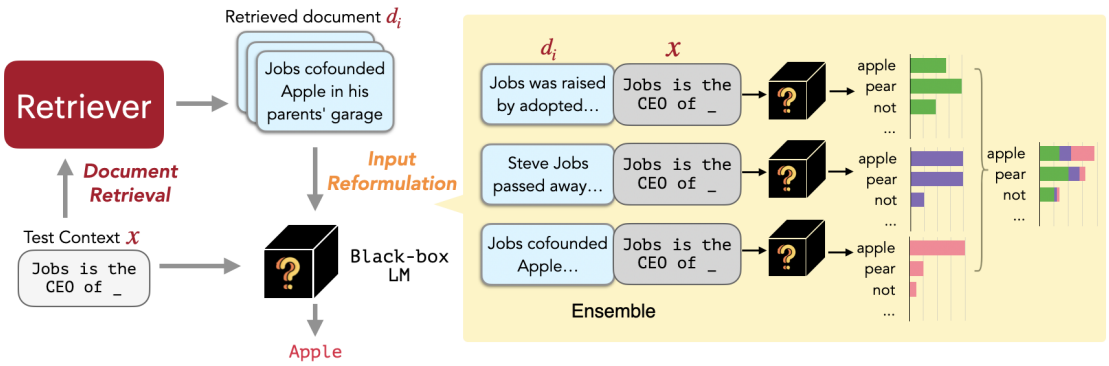

Figure 3.1: REPLUG inference. Each retrieved document is independently prepended to the input and processed by the frozen black-box LM; the resulting distributions are ensembled using retrieval scores as weights. The LM is never modified.

3.1 REPLUG: Retrieval Augmentation for Black-Box LMs

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich 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*.

3.1.1 Motivation and Problem Setting

Prior work on retrieval-augmented LMs [Borgeaud et al., 2022; Izacard and Grave, 2021] requires finetuning LMs further or modifying the language model's architecture or parameters to integrate retrieved content. RETRO [Borgeaud et al., 2022] adds new cross-attention layers specifically designed to attend to retrieved chunks, requiring architectural changes and complete retraining from scratch. Fusion-in-Decoder [Izacard and Grave, 2021] encodes retrieved passages with an encoder and fuses them in the decoder.

These requirements preclude applying retrieval augmentation to the largest and most capable LMs, which are typically accessible only through API endpoints: users can submit a text prompt and receive a generation or log-probability scores, but cannot access or modify internal representations or parameters. In 2022, when this work was developed, models such as GPT-3 (175B) and Codex were available only as black-box APIs. Demonstrating that retrieval augmentation is beneficial for these models was an open and important question.

3.1.2 The REPLUG Framework

We introduce **REPLUG** [Shi et al., 2024d] (Retrieve and Plug), a retrieval-augmented LM framework that treats the language model as a black box and augments it with a tunable retrieval module.

Inference. Given an input context c, REPLUG first retrieves the top-k documents $\{d_1, \ldots, d_k\}$ from an external corpus $\mathcal{D}$ using an off-the-shelf retriever:

$$\text{score}(d_i, c) = \text{sim}(\text{enc}(d_i), \text{enc}(c)), \tag{3.1}$$

where enc is a dense encoder and sim is cosine similarity. Each retrieved document is prepended to the context to form a new input $d_i \circ c$, which is passed independently through the black-box LM. The LM's next-token probability distributions for each retrieved document are then ensembled:

$$p(y \mid c) = \sum_{d \in \mathcal{D}'} \lambda(d, c) \cdot p_{\text{LM}}(y \mid d \circ c), \tag{3.2}$$

where $\lambda(d, c) = e^{s(d,c)} / \sum_{d' \in \mathcal{D}'} e^{s(d',c)}$ is the retrieval weight (softmax-normalized similarity score). This ensemble scheme allows REPLUG to incorporate multiple retrieved documents without exceeding the LM's context window: each document is processed in parallel rather than concatenated.

LM-Supervised Retrieval Training (REPLUG LSR). We further introduce a self-supervised training algorithm that adapts the retriever to the target LM. Given input context x and ground-truth continuation y, we define two distributions over the retrieved documents $\mathcal{D}'$. The *retrieval distribution* is computed from similarity scores:

$$P_R(d \mid x) = \frac{e^{s(d,x)/\gamma}}{\sum_{d' \in \mathcal{D}'} e^{s(d',x)/\gamma}}, \tag{3.3}$$

where γ is a temperature hyperparameter. The *LM-scored distribution* uses the frozen LM's likelihood of the ground-truth output given each document as a relevance signal:

$$Q(d \mid x, y) = \frac{e^{P_{\text{LM}}(y|d \circ x)/\beta}}{\sum_{d' \in \mathcal{D}'} e^{P_{\text{LM}}(y|d' \circ x)/\beta}}, \tag{3.4}$$

where β is a temperature hyperparameter. The retriever is trained by minimizing the KL divergence between these two distributions:

$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \mathrm{KL}\Big(Q(d \mid x, y) \parallel P_R(d \mid x)\Big), \tag{3.5}$$

where $\mathcal{B}$ is a batch of training contexts. Only the retriever parameters are updated; the LM remains frozen, requiring only API-level access.

3.1.3 Results

REPLUG demonstrates that retrieval augmentation benefits large-scale black-box LMs across diverse settings. Without any retrieval training:

- Augmenting GPT-3 (175B) with off-the-shelf Contriever retrieval reduces language modeling perplexity on the Pile by 3.8% relative.
- Augmenting Codex [Chen et al., 2021] (175B) with retrieval improves performance on MMLU [Hendrycks et al., 2021] by 4.5%, achieving performance comparable to the 540B instruction-finetuned Flan-PaLM [Chung et al., 2022] on this benchmark.

With LSR retriever training:

- REPLUG LSR reduces GPT-3 (175B) language modeling perplexity by 6.3% relative, significantly better than any off-the-shelf retriever.
- The trained retriever learns to retrieve documents that are specifically helpful for the target LM, rather than merely semantically similar to the query.

A qualitative analysis of the documents retrieved by the LSR-trained retriever versus off-the-shelf Contriever, together with an inference-cost analysis as a function of the number of retrieved documents, is provided in Appendix C (Tables C.1 and C.2).

REPLUG establishes that retrieval augmentation benefits very large-scale black-box LMs, and that a retriever adapted to the LM's own preferences outperforms off-the-shelf alternatives, both without modifying the LM.

Table 3.1: Both REPLUG and REPLUG LSR consistently enhance different language models. Bits per byte (BPB) on the Pile for GPT-2 and GPT-3 families. Gain % = relative improvement over the original model.

Model		# Params	Original	+REPLUG	Gain%	+REPLUG LSR	Gain%
GPT-2	Small	117M	1.33	1.26	5.3	1.21	9.0
	Medium	345M	1.20	1.14	5.0	1.11	7.5
	Large	774M	1.19	1.15	3.4	1.09	8.4
	XL	1.5B	1.16	1.09	6.0	1.07	7.8
GPT-3	Ada	350M	1.05	0.98	6.7	0.96	8.6
(black-box)	Babbage	1.3B	0.95	0.90	5.3	0.88	7.4
	Curie	6.7B	0.88	0.85	3.4	0.82	6.8
	Davinci	175B	0.80	0.77	3.8	0.75	6.3

3.2 In-Context Pretraining

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Wen-tau Yih, and Mike Lewis. 2024. In-Context Pretraining: Language Modeling Beyond Document Boundaries. In *International Conference on Learning Representations* (Spotlight).

3.2.1 Motivation

Standard LM pretraining concatenates documents from the training corpus sequentially, often padding or truncating to fill fixed-length sequences. The resulting training examples mix documents in arbitrary order, and the model sees each document in the context of whatever happens to precede it in the batch. This has an important implication: the model is never trained to reason across *related* documents, to synthesize information from multiple sources that together address a common topic or query.

At inference time with retrieval augmentation, however, the model must do exactly this: integrate information from multiple retrieved documents that are topically related to the query. The mismatch between pretraining (random document concatenation) and retrieval-augmented inference (related document concatenation) is a fundamental source of underperformance in retrieval-augmented systems.

3.2.2 The In-Context Pretraining Algorithm

We introduce **In-Context Pretraining** (ICLM), which reorganizes the pretraining document sequence to expose the model to topically related documents during training. The key steps are:

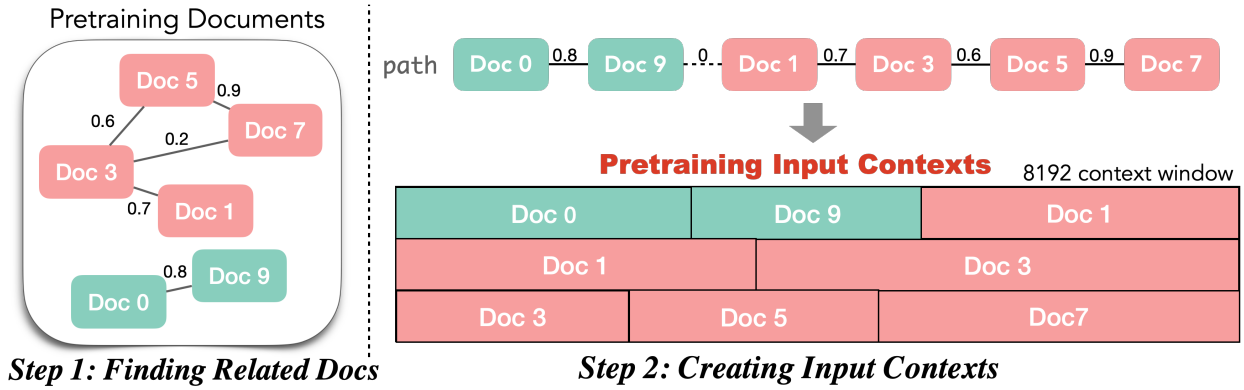

Figure 3.2: In-Context Pretraining (ICLM) reorganizes pretraining sequences. Standard pretraining concatenates randomly ordered documents; ICLM builds a retrieval index, finds topically related documents for each training document, and concatenates them into a single training sequence. This trains the LM to reason across related documents, directly matching retrieval-augmented inference.

1. **Build a retrieval index.** We build a dense retrieval index over the entire pretraining corpus.

2. **Find related documents.** For each document d in the corpus, we retrieve its top-k most similar documents $\{d_1^*, \ldots, d_k^*\}$ from the corpus.

3. **Construct related-document sequences.** We form training sequences by concatenating d with its retrieved neighbors $\{d_1^*, \ldots, d_k^*\}$, ordered by relevance.

4. **Pretrain on related sequences.** The LM is pretrained on these related-document sequences, with standard next-token prediction loss.

This simple reorganization teaches the LM to synthesize information across related documents, directly matching the inference-time setting in retrieval-augmented generation.

3.2.3 Results

We pretrain LLaMA-architecture models at four scales (0.3B, 0.7B, 1.5B, and 7B parameters) on 306 billion tokens of English CommonCrawl, comparing ICLM to standard random-shuffle pretraining and to the KNN-pretraining baseline that naively concatenates each document with its retrieved neighbors.

Language modeling. ICLM outperforms both standard pretraining and KNN on held-out language modeling perplexity across Wikipedia, Arxiv, and Books corpora at every model scale. The gains are consistent or grow larger with model size, indicating that the richer pretraining signal from related-document contexts becomes more valuable as capacity increases.

In-context learning. On seven text classification benchmarks spanning sentiment analysis, topic classification, and hate speech detection, ICLM achieves an **8% average gain** over standard pretraining in the 32-shot in-context learning setting. Training on topically coherent document sequences thus improves the model's ability to learn from in-context demonstrations.

Reading comprehension. ICLM outperforms standard pretraining by an average of **14%** across five reading comprehension benchmarks (RACE-High, RACE-Middle, SQuAD, BoolQ, HotpotQA). The improvement is largest on HotpotQA, which requires integrating evidence across multiple documents, precisely the setting that ICLM's training directly addresses.

Retrieval-augmented question answering. On Natural Questions and TriviaQA in the open-book setting (top-10 retrieved Wikipedia documents as context), ICLM outperforms standard pretraining by **9%**. The KNN-pretraining baseline, despite using the same training objective as retrieval-augmented inference, fails to improve over standard pretraining here, because its document-repetition problem causes overfitting to a small subset of the corpus. ICLM's graph-based document ordering avoids this by ensuring each document appears exactly once.

Factuality. On knowledge-conflict benchmarks (NQ-Swap [Longpre et al., 2021] and MemoTrap [Liu and Liu, 2023]) where the provided context contradicts the model's parametric beliefs, ICLM outperforms standard pretraining substantially. Training on related documents appears to develop stronger habits of context-following, complementing the Context-Aware Decoding method (§3.3) that we develop separately to address this problem at inference time.

Table 3.2: ICLM outperforms standard pretraining on 32-shot in-context learning across seven text classification datasets. All models are 7B-scale LLaMA-architecture models trained on 306B tokens.

Method	Sentiment			Hate Speech		Topic		Avg.
	Amazon	SST2	Yelp	Hate	Offensive	Agnews	DBpedia	
Standard	94.6	83.7	74.3	52.7	55.7	68.3	61.5	66.0
kNN-pretrain	88.0	80.2	65.1	50.1	53.1	65.7	56.4	61.8
ICLM (ours)	**96.5**	**93.2**	**77.4**	**60.6**	**57.3**	**76.0**	**63.2**	**71.3**

3.3 Context-Aware Decoding

Weijia Shi*, Xiaochuang Han*, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting Your Evidence: Hallucinate Less with Context-Aware Decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*.

3.3.1 Motivation

Even when relevant context is provided in the LM's input (either through retrieval augmentation or as part of the prompt), LMs often fail to faithfully attend to it. A common phenomenon is that LMs can be overconfident in their prior parametric knowledge: when asked a question in the presence of a document that contradicts the LM's trained beliefs, the model may still produce an answer based on its prior knowledge rather than the provided evidence [Longpre et al., 2021; Zhou et al., 2023].

For example, when LLaMA is presented with the document "Argentina won the FIFA World Cups in 1978, 1986 and 2022" in its context and asked "How many World Cups has Argentina won?", the model may respond "Two" based on its pretraining data (which predates the 2022 World Cup) rather than the provided context.

3.3.2 Context-Aware Decoding

We introduce **Context-Aware Decoding** (CAD), a training-free decoding method that amplifies the LM's attention to contextual information during generation.

The key idea is that we run the LM twice, once with the context document and once without (input

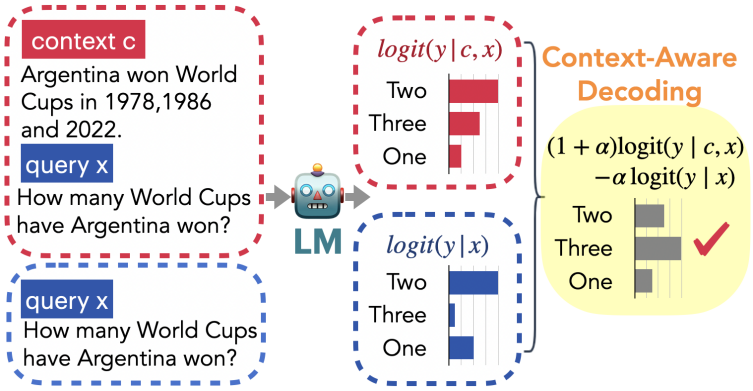

Figure 3.3: Context-Aware Decoding (CAD) amplifies the LM's attention to provided context. The LM is run twice, once with and once without the context document, and the output distributions are contrasted. Tokens favored by the context over the prior receive boosted probability, encouraging the model to follow retrieved evidence over parametric memory.

prompt only), and amplify the difference between the two output distributions:

$$p_{\text{CAD}}(x \mid c, d) \propto p_{\text{LM}}(x \mid c, d)^{1+\alpha} \cdot p_{\text{LM}}(x \mid c)^{-\alpha}, \tag{3.6}$$

where $\alpha \geq 0$ is a hyperparameter controlling the strength of context amplification, c is the input query/context, and d is the retrieved or provided document. When $\alpha = 0$, this reduces to standard decoding. When $\alpha > 0$, tokens that are more likely given the document than without it receive increased probability, and tokens that are likely without the document receive decreased probability.

3.3.3 Results

We evaluate CAD on:

- **Summarization**: CNN/DailyMail and XSum, measuring faithfulness of summaries to their source documents. CAD applied to LLaMA-30B improves ROUGE-L by 21% and factuality scores by 14.3%.
- **Knowledge conflict QA**: NQ-Swap [Longpre et al., 2021], where the provided context contradicts the model's prior knowledge. CAD brings a 2.9× improvement for LLaMA-30B in correctly following the contextual evidence.
- **Instruction following**: CAD helps instruction-tuned models adhere to detailed output constraints

Table 3.3: CAD substantially improves faithfulness to retrieved context on NQ-SWAP (knowledge conflict) and standard NQ, across four model families. Memo. = memorization score; higher NQ-SWAP = better context following.

Model		Decoding	Memo.	NQ	NQ-SWAP
OPT	13B	Regular	32.5	29.2	18.8
		CAD	**44.5**	**32.2**	**36.9**
	30B	Regular	28.4	29.4	14.7
		CAD	**41.0**	**35.5**	**29.0**
GPT-J	3B	Regular	22.5	31.9	19.1
		CAD	**47.3**	**39.9**	**41.2**
	20B	Regular	37.1	22.8	16.1
		CAD	**57.3**	**32.1**	**36.8**
LLaMA	13B	Regular	23.8	22.3	11.7
		CAD	**57.1**	**33.6**	**36.7**
	30B	Regular	25.8	23.8	9.6
		CAD	**50.6**	**34.0**	**37.7**
FLAN	3B	Regular	69.2	**81.8**	71.4
		CAD	**72.2**	80.3	**73.3**
	11B	Regular	82.0	**85.5**	73.0
		CAD	**88.7**	82.5	**77.1**

specified in the prompt.

The benefit of CAD grows with model scale on knowledge-conflict tasks: larger models hold their prior beliefs more strongly, and so gain more from the contrastive mechanism. CAD is directly applicable to any black-box LM that provides log-probability scores.

3.4 Related Work

Retrieval-augmented language models. The idea of combining parametric language models with non-parametric text stores has a long history. kNN-LM [Khandelwal et al., 2020] interpolates an LM's next-token distribution with a nearest-neighbor distribution over a datastore of training contexts, without modifying the LM. REALM [Guu et al., 2020] and RAG [Lewis et al., 2020] train end-to-end retrievers that supply documents to encoder-decoder models. RETRO [Borgeaud et al., 2022] adds dedicated chunked cross-attention layers within the transformer architecture. These approaches either require access to model internals or architectural changes, making them inapplicable to large API-only models. REPLUG is the first framework for retrieval augmentation of completely black-box LMs. Concurrent and subsequent work has extended

REPLUG's ideas to instruction-tuned models [Lin et al., 2024] and to joint retriever-generator training [Lin et al., 2024].

Dense retrieval and embedding models. DPR [Karpukhin et al., 2020] established the dominant paradigm for dense passage retrieval using bi-encoder models trained on question-answer pairs. Contriever [Izacard et al., 2022] showed that contrastive self-supervised pretraining without human annotation can produce competitive retrievers. GTR [Ni et al., 2022] demonstrated that scaling the dual encoder architecture dramatically improves generalization. INSTRUCTOR differs from all prior embedding models by conditioning on task instructions, enabling a single model to serve as a universal embedder across arbitrary downstream tasks. Concurrent instruction-following embedding work [Su et al., 2023] independently proposed similar ideas.

Compression and selective augmentation. The computational overhead of retrieval augmentation has motivated compression-based approaches. FILCO [Wang et al., 2023] filters retrieved sentences using learned classifiers. LLMLingua [Jiang et al., 2023] compresses prompts using a small language model as a proxy. RECOMP is the first work to jointly address compression and selective augmentation, training compressors directly with LM-derived supervision and training a gating mechanism to skip retrieval entirely when unnecessary.

Faithful context utilization. The problem of LMs ignoring provided context has been studied as "sycophancy" [Perez et al., 2022] and "knowledge conflict" [Longpre et al., 2021]. Contrastive decoding [Li et al., 2023b] was originally proposed to contrast a large LM against a small one to improve generation quality. CAD adapts this idea to contrast generation with versus without the retrieved document, specifically targeting context faithfulness. Subsequent work has extended context-aware contrastive decoding to instruction following [Shi et al., 2024b] and multi-document settings.

3.5 Conclusion

This chapter collectively addresses the key components of a retrieval-augmented language model pipeline: augmenting black-box LMs with retrieval (REPLUG), training better embeddings for retrieval (INSTRUCTOR),

improving the LM's ability to use retrieved information through better pretraining (ICLM), compressing retrieved content for efficiency (RECOMP), and ensuring the model faithfully attends to retrieved context (CAD). Together, they demonstrate that model modularity through augmentation is a rich research area spanning retrieval models, pretraining algorithms, compression, selective augmentation, and decoding.

Chapter 4

Model Modularity: Multimodal Augmentation

The previous chapter demonstrated that external text retrievers can supply knowledge to LMs, substantially improving factuality. This chapter extends the modular principle from text to the visual domain. Multimodal language models face a version of the same problem as text-only LMs: they must compress an enormous range of visual knowledge into fixed parameters, leading to hallucination on rare visual concepts and on tasks requiring precise spatial reasoning.

We pursue this in two complementary directions. We first briefly describe RA-CM3 [Yasunaga et al., 2023], a precursor contribution that extends retrieval augmentation to multimodal generation, and then turn to the main focus of this chapter, Visual Sketchpad (§4.1).

RA-CM3 is the first retrieval-augmented model capable of generating arbitrary interleaved sequences of text and images. It pairs a CLIP-based [Radford et al., 2021] mixed-modal retriever with a multimodal generator built on CM3 [Aghajanyan et al., 2022], a Transformer decoder that tokenizes images into discrete tokens so text and images share a single sequence. Given a query, the retriever fetches relevant image-text documents from an external memory; these are prepended as in-context references so the generator can ground its outputs in concrete examples rather than relying solely on parametric knowledge. Only the generator is trained, while the retriever stays frozen, and a small auxiliary loss on the retrieved documents recycles the otherwise-wasted computation spent encoding their image tokens. This grounding improves both quality

and compute efficiency: on MS-COCO, RA-CM3 attains an FID of 16 for caption-to-image generation (versus 28 for DALL-E, despite DALL-E being 3× larger) and a CIDEr of 89 for zero-shot captioning, using less training compute than comparable autoregressive systems. Ablations further show that retrieving full multimodal documents and enforcing diversity among them are both essential. RA-CM3 thus established the first systematic evidence that retrieval augmentation benefits joint multimodal generation; we discuss it here and in the related work (§4.2) rather than in a dedicated section.

The main focus of this chapter is Visual Sketchpad (§4.1), a framework that goes beyond retrieving static content: the LM dynamically invokes specialist vision models to create visual artifacts, annotated images, depth maps, segmentation masks, that serve as intermediate steps in its reasoning process, enabling a visual chain-of-thought.

A key observation motivating Visual Sketchpad is that text-only chain-of-thought [Wei et al., 2022] is fundamentally inexpressive for spatial and relational reasoning. Just as humans sketch when solving geometry problems or annotate diagrams when analyzing circuits, LMs benefit from maintaining state in the visual domain rather than forcing all reasoning through language.

4.1 Visual Sketchpad: Sketching as a Visual Chain of Thought

Yushi Hu*, Weijia Shi*, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. 2024. Visual Sketchpad: Sketching as a Visual Chain of Thought for Multimodal Language Models. In *Advances in Neural Information Processing Systems*.

4.1.1 Motivation

When humans solve complex visual or mathematical problems, they frequently sketch. A student solving a geometry problem draws auxiliary lines. An engineer analyzing a circuit diagram annotates components. A scientist examining a microscope image marks regions of interest. These visual artifacts are not just communication tools, they are reasoning tools, externalizing intermediate computations and making spatial relationships explicit.

Current multimodal LMs do not have this capability. Their chain-of-thought reasoning, even when applied to image inputs, is entirely textual [Wei et al., 2022]. This is a fundamental limitation for tasks

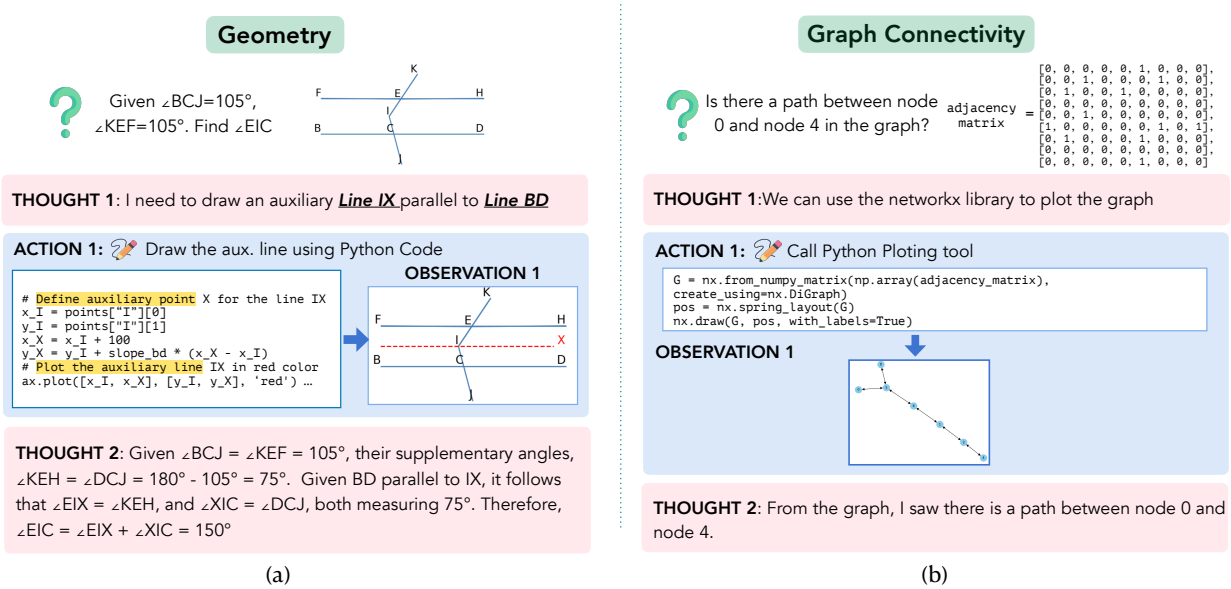

Figure 4.1: Visual Sketchpad enables multimodal LMs to reason step-by-step across modalities. The LM can invoke specialist vision models (e.g., object detectors, depth estimators) to create visual annotations, which become part of the visual reasoning context for subsequent steps.

involving spatial reasoning, geometric computation, or fine-grained visual perception: text-based intermediate representations are simply inexpressive for capturing the spatial and relational structure of visual scenes.

Recent work has explored tool-augmented reasoning for vision [Gupta and Kembhavi, 2023; Surís et al., 2023], where LMs generate code that invokes vision tools in sequence. However, the outputs of these tools are invariably returned as text (bounding box coordinates, classification labels, etc.), and the LM's reasoning remains in the text domain. No intermediate visual state is maintained.

4.1.2 The Visual Sketchpad Framework

We introduce **Visual Sketchpad**, a framework that equips multimodal LMs with the ability to create and reason with visual artifacts as part of their chain of thought. The key idea is to allow the LM to invoke vision specialist models that return *images*, not text descriptions, as outputs, which are then incorporated back into the LM's visual context for subsequent reasoning steps.

Reasoning with sketches. At each reasoning step, the LM can invoke any of a suite of specialist vision models:

- **Drawing tools**: bounding boxes, auxiliary lines, points, and arrows that highlight spatial relationships.
- **Perception tools**: object detection (DETR), depth estimation, instance segmentation (SAM), optical flow, and edge detection.
- **Analytical tools**: geometric calculators (line lengths, angles), Python code execution for numerical computation.

Each tool returns a visual output, an annotated image or a new image layer, that is added to the LM's visual context. The LM then reasons over the updated visual context to decide the next step or to produce the final answer. This creates a visual chain of thought where intermediate computations are expressed and reasoned over in the image domain rather than being forced into text.

Implementation. SKETCHPAD is implemented as a multi-turn interaction protocol:

1. The LM receives the original image and problem statement.
2. The LM generates a reasoning step, which may include invoking a vision tool (expressed as a structured tool call).
3. The tool executes and returns a visual output, which is added to the LM's image context.
4. Steps 2-3 repeat until the LM produces a final answer.

This design requires no modification to the underlying LM, it uses the model's existing multimodal input and tool-calling capabilities. SKETCHPAD is evaluated using GPT-4o [OpenAI, 2023] as the backbone LM.

4.1.3 Evaluation Tasks

We evaluate SKETCHPAD on two categories of tasks where visual reasoning is essential:

Mathematical reasoning with visual inputs.

- **Geometry3K**: geometry problem solving requiring understanding of angles, line lengths, and spatial relationships.

- **MathVista**: visual mathematical reasoning spanning charts, diagrams, and geometric figures.
- **FunctionQA**: reasoning about function graphs.

Visual perception tasks.

- **BLINK** [Fu et al., 2024]: a benchmark of 14 visual perception tasks that current multimodal LMs struggle with, including relative depth, visual correspondence, and multi-view reasoning.
- **Vision benchmarks**: spatial reasoning, counting, and object localization tasks.

4.1.4 Results

SKETCHPAD achieves strong performance improvements across evaluated tasks:

Mathematical reasoning. Applied to GPT-4o, SKETCHPAD achieves an average absolute accuracy improvement of **12.7%** across mathematical reasoning benchmarks compared to GPT-4o with text-only chain-of-thought. The improvement is largest for geometry tasks (where drawing auxiliary lines and marking angles is most helpful) and for function reasoning (where plotting intermediate computations reveals structure invisible in the raw formula).

Visual perception. On visual perception tasks from BLINK, SKETCHPAD achieves an average absolute improvement of **8.6%** across tasks. The improvements are concentrated in tasks that require understanding spatial relationships (relative depth, visual correspondence, multi-view reasoning), where the ability to invoke depth estimators, segmentation models, and optical flow tools provides crucial information that text-only reasoning cannot express.

New state-of-the-art. GPT-4o with SKETCHPAD sets new state-of-the-art results on multiple benchmarks, demonstrating that equipping existing frontier models with visual reasoning tools can be more effective than developing new multimodal architectures trained from scratch for these tasks.

Table 4.1: SKETCHPAD establishes new state-of-the-art on all complex visual reasoning tasks. Accuracy (%) for GPT-4 Turbo and GPT-4o with and without SKETCHPAD. **+gains** shown in green.

Model	V^*Bench	MMVP	Depth	Spatial	Jigsaw	Vis. Corr.	Sem. Corr.
Prior multimodal LMs							
LLaVA-1.5-7B	48.7		52.4	61.5	11.3	25.6	23.0
LLaVA-1.5-13B		24.7	53.2	67.8	58.0	29.1	32.4
Gemini-Pro	48.2	40.7	40.3	74.8	57.3	42.4	26.6
GPT-4V	55.0	38.7	59.7	72.7	70.0	33.7	28.8
Previous SOTA	75.4	49.3	67.7	76.2	70.0	42.4	33.1
+ *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**

4.1.5 Analysis

Why do visual sketches help? We analyze which tool invocations contribute most to performance improvements. On geometry tasks, auxiliary lines and angle markers are the most valuable sketching actions, as they make implicit spatial relationships explicit. On depth and correspondence tasks, depth maps and segmentation masks provide grounded visual representations that the LM can reason over directly. A per-category breakdown of tool-invocation frequency, along with an analysis of common failure modes, is reported in Appendix D (Tables D.1 and D.2).

Generalization of the visual chain of thought. A key finding is that SKETCHPAD improves performance on tasks *not specifically targeted* by any single tool, the visual chain of thought generalizes beyond the individual tool capabilities. This suggests that the framework benefits the LM by providing visual grounding, even when the specific visual artifacts do not directly solve the problem.

4.2 Related Work

Vision-language models. A large body of work has developed models for multimodal understanding, primarily by adapting pretrained language models to process image inputs. Flamingo [Alayrac et al., 2022]

uses cross-attention from text to visual features for few-shot visual question answering. BLIP-2 [Li et al., 2023a] uses a Q-Former module to bridge a frozen vision encoder and a frozen LM. LLaVA [Liu et al., 2023] fine-tunes an LM on instruction-following image-text data using a simple linear projection from visual to language space. These models excel at tasks where the answer can be expressed in text given a static image, but they do not support visual reasoning or generation.

Retrieval-augmented multimodal generation. Re-Imagen [Chen et al., 2023a] retrieves images to condition a cascaded diffusion model, improving faithful generation of rare entities. KNN-Diffusion retrieves image neighbors for diffusion models at inference time. Our work [Yasunaga et al., 2023] differs in jointly supporting text and image generation within a single autoregressive model, and in studying the scaling laws of retrieval for multimodal models.

Tool-augmented reasoning. Visual Programming [Gupta and Kembhavi, 2023] and ViperGPT [Surís et al., 2023] augment language models with Python code execution that invokes vision APIs. These systems return tool outputs as text (bounding box coordinates, labels), keeping all reasoning in the language domain. SKETCHPAD differs fundamentally by returning tool outputs as *images*, enabling visual chain-of-thought reasoning. Concurrent work on multimodal reasoning chains (e.g., PoT [Chen et al., 2023b]) focuses on text-based intermediate computations. Our visual chain-of-thought framework connects to subsequent work on thinking with images in models like o3.

Visual reasoning benchmarks. BLINK [Fu et al., 2024] was specifically designed to evaluate visual perception tasks where current models struggle. MathVista [Lu et al., 2024b] provides a comprehensive benchmark of mathematical reasoning with visual inputs. SKETCHPAD's strong improvements on both families of benchmarks validate the benefit of visual intermediate steps for tasks requiring spatial reasoning and geometric computation.

4.3 Conclusion

This chapter extends model modularity from the text domain to the multimodal domain, demonstrating that augmenting LMs with specialized external modules generalizes naturally across modalities.

Retrieval-augmented multimodal generation [Yasunaga et al., 2023] demonstrated that supplementing a multimodal model's parametric knowledge with retrieved image-text evidence improves both generation quality and compute efficiency. Visual Sketchpad [Hu et al., 2024] extended this further: instead of retrieving static content, the LM dynamically invokes specialist perception models that *transform* the visual input, creating new visual representations that serve as intermediate reasoning steps. The accuracy improvements of SKETCHPAD on GPT-4o confirm that visual reasoning remains a bottleneck even for the most capable frontier models, and that external specialist tools can address that bottleneck without any fine-tuning of the base model.

4.4 Limitations

Dependence on frontier model API. SKETCHPAD was evaluated using GPT-4o as the backbone and relies on reliable, capable tool-calling APIs. Its applicability to smaller, open-weight models is not validated comprehensively; these models may struggle to consistently generate well-formed tool calls or to reason over multi-step visual contexts. Subsequent work has begun to address this gap: Visual-ARFT [Liu et al., 2025] extends visual agentic tool use to open-weight models such as Qwen2.5-VL-7B via reinforcement fine-tuning, demonstrating that the core capabilities of visual tool invocation and image-based reasoning can be trained into smaller models without relying on proprietary APIs.

Tool coverage and selection. The specialist tools available in SKETCHPAD cover a useful but finite set of visual operations. Tasks requiring tool types not in the toolkit (e.g., 3D understanding, thermal imaging) cannot benefit without adding new tools, which requires engineering effort and API access to appropriate models. More broadly, an open question is how to discover which tools are most valuable for a given task domain, or how to train the model to invoke tools more strategically, for example, through reinforcement learning on tool invocation sequences.

Retrieval-augmented generation latency. The retrieval-augmented multimodal generation system requires retrieving multiple image-text documents at inference time, which adds latency proportional to the number of retrieved documents and the size of the retrieval corpus. Compression and selective augmentation strategies

analogous to RECOMP are understudied for the multimodal setting.

Chapter 5

Data Modularity: Decentralized and Composable Language Models

The previous two chapters focus on model modularity, augmenting language models with external components at inference time to overcome knowledge limitations and improve visual reasoning. However, the privacy and copyright issues documented in Chapter 2 are baked in during pretraining: by the time the model is deployed, the sensitive data has already been memorized into its parameters.

This chapter addresses a more fundamental question: must the language model itself be trained monolithically on all data? We argue the answer is no, and introduce architectures and training algorithms that build LMs from decentralized, independently trained components. We call this paradigm *data modularity*, following the vision [Raffel, 2023] of building machine learning models like open source software: just as software packages are developed independently and composed into systems, data-modular LMs are built from expert components trained on separate data that can be flexibly combined at inference time. Rather than centralizing all data and training a single monolithic model, data-modular models train separate expert components on different data subsets and compose them at inference time. This construction provides data control that machine unlearning cannot: rather than attempting to remove information from a model that has already absorbed it, data-modular models keep each data owner's contribution cleanly separable as a distinct component that can be included or excluded by design.

Data modularity offers several advantages over the monolithic paradigm:

- **Privacy**: data owners can contribute to model development without sharing their data with a central model developer.
- **Flexibility**: specific data sources can be included or excluded from the deployed model at inference time, enabling fine-grained control.
- **Efficiency**: modality-specific or domain-specific components can be developed in parallel, reusing existing pretrained models rather than training from scratch.
- **Updatability**: new data sources can be incorporated by training and merging a new expert module, without retraining the entire model.

We present two contributions: FLEXOLMO for privacy-preserving collaborative training via mixture-of-experts (§5.1), and LLAMAFUSION for efficient multimodal model development from pretrained components (§5.2).

5.1 FlexOlmo: Collaborative Training on Private Datasets

Weijia Shi, Akshita Bhagia, Kevin Farhat, Niklas Muennighoff, Pete Walsh, Jacob Morrison, Dustin Schwenk, Shayne Longpre, Jake Poznanski, Allyson Ettinger, and others. 2025. FlexOlmo: Open Language Models for Flexible Data Use. In *Advances in Neural Information Processing Systems* (Spotlight).

5.1.1 Motivation

Standard LM pretraining requires all training data to be centralized. This centralization is a fundamental barrier to using data from organizations or individuals who cannot share their data for regulatory, competitive, or confidentiality reasons. The scale of this limitation is significant: a lot of high-quality data, medical records, legal documents, and proprietary scientific data, are locked away from LM developers due to privacy constraints.

Federated learning [Konečnỳ et al., 2016; McMahan et al., 2017; Kairouz et al., 2021] offers an approach to training without centralizing data, but practical adoption has been limited. Federated training of LMs requires synchronized gradient communication across participants, which is both technically complex and

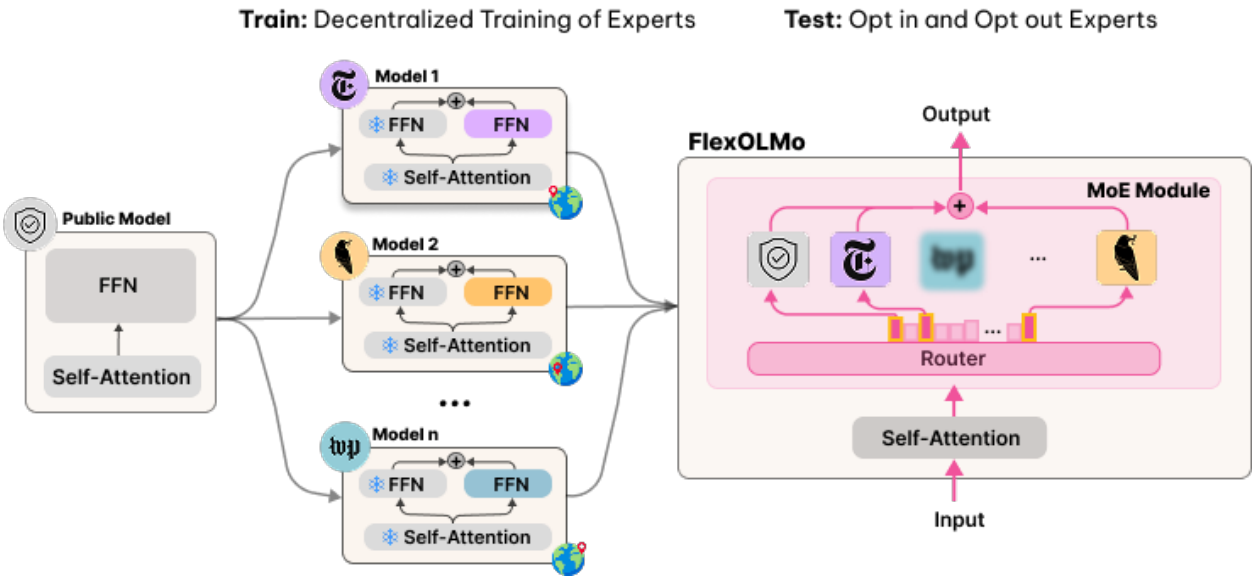

Figure 5.1: Overview of FLEXOLMO. Each data owner independently trains an expert FFN module anchored to a shared public model. At inference time, experts are merged into a single MoE via router embedding concatenation, and any expert can be included or excluded without retraining.

computationally expensive. Performance degradation from communication bottlenecks and data heterogeneity is substantial [Wei et al., 2019; Zhang et al., 2022b].

We seek a simpler alternative: can data owners independently train components on their private data and subsequently merge these components into a single capable model, without any synchronized training?

5.1.2 The FlexOlmo Framework

We introduce **FLEXOLMO** [Shi et al., 2025a], a mixture-of-experts LM architecture that enables fully asynchronous, distributed training on private datasets with flexible opt-in and opt-out at inference time.

Architecture. FLEXOLMO replaces the FFN in each transformer block with a sparse mixture-of-experts layer. Given an input token embedding $\mathbf{x} \in \mathbb{R}^h$, the MoE layer computes:

$$\mathbf{y} = \sum_{i \in \text{Top-}k(r(\mathbf{x}))} \text{softmax}(r(\mathbf{x})_i)\, M_i(\mathbf{x}), \tag{5.1}$$

where $r(\mathbf{x}) = \mathbf{W}_r\mathbf{x}$ is the router and $\{M_i\}$ are the expert FFNs. Unlike standard MoEs where all experts and the router are trained jointly, FLEXOLMO trains each expert M_i independently on a separate private dataset

D_i.

Training: Coordinated expert learning. The central challenge is training experts independently such that they can be merged without any joint training. We use the public model M_{pub} as a shared anchor. For each private dataset D_i, we construct a two-expert mini-MoE with both experts initialized from M_{pub}. During training on D_i, the public expert and all attention layers are frozen, while only the private expert M_i is updated. This ensures each expert learns to complement the public model rather than replace it; since all experts coordinate with the same anchor, they implicitly coordinate with each other at merge time.

Domain-informed router. We avoid joint router training with a **router embedding concatenation** approach. The router matrix $\mathbf{W}_r$ is decomposed into per-expert row embeddings, where each $\mathbf{r}_i$ is initialized as the average embedding of a sample S_i from D_i:

$$\mathbf{r}_i = \frac{1}{|S_i|} \sum_{d_k \in S_i} \mathbf{E}(d_k), \tag{5.2}$$

using an off-the-shelf text embedder $\mathbf{E}$. Each $\mathbf{r}_i$ is fine-tuned alongside M_i during expert training. At merge time, $\mathbf{W}_r$ is formed by stacking all expert embeddings — no additional joint training required.

Negative bias for multi-expert competition. Since each expert is trained only in a pairwise setup against the public model, it never directly competes with other private experts during training. To address this at inference time, we add a learned negative bias b_i per private expert: M_i is only selected when $\mathbf{r}_i \cdot \mathbf{x} + b_i > \mathbf{r}_{\text{pub}} \cdot \mathbf{x}$, otherwise defaulting to the public expert. This prevents over-activation of private experts on out-of-domain inputs.

Opt-in and opt-out. Any expert can be included or excluded at inference time by simply adding or removing its row from $\mathbf{W}_r$ and renormalizing. This provides an exact, guaranteed opt-out mechanism: the data owner's contribution is entirely encapsulated in a single expert module that can be switched off with no effect on the rest of the model.

5.1.3 Experimental Setup

The public model M_{pub} is a 7B-parameter dense LM following the OLMo-2 architecture, trained on 1 trillion tokens of Common Crawl. Seven private expert modules are then independently trained, each for 50B tokens, on domain-specific datasets representing data that real organizations may be unwilling or unable to share: news, creative writing, code, academic papers, educational text, math, and Reddit. The final FLEXOLMO model has 37B total parameters with 20B active (top-4 of 8 experts per token), and is evaluated on 31 tasks spanning 10 categories.

5.1.4 Results

FlexOlmo outperforms individual experts. Merging all experts improves upon the public-only model by **41% on average** across 31 tasks. Gains are especially large on domain-specific benchmarks: BBH ($35.6 \rightarrow 47.1$), math ($8.1 \rightarrow 50.7$), and coding ($1.0 \rightarrow 17.3$). The merged model matches or exceeds individual domain experts even on their own tasks, demonstrating synergistic cross-domain benefits that no single expert achieves alone.

Outperforms all merging baselines. FLEXOLMO beats the strongest baseline (BTM) by **10% relative** on average, and outperforms model soup and output ensembling by 10.1% each. The advantage comes from sparse token-level routing: each token is processed by only the most relevant experts, avoiding the cross-domain interference that affects parameter averaging and output ensembling.

Routing behavior. As shown in Figure 5.2, the router routes domain-matched inputs to the corresponding expert (e.g., math inputs activate the math expert). The public expert is frequently co-activated, consistent with the coordinated training design. Different expert combinations are activated at different transformer layers, giving the model expressivity beyond prompt-based routing approaches that assign a single expert per input.

Data opt-out. Removing any expert at inference time reduces performance only on that expert's in-domain tasks while leaving unrelated tasks largely unaffected, confirming clean, targeted opt-out without collateral degradation across other domains.

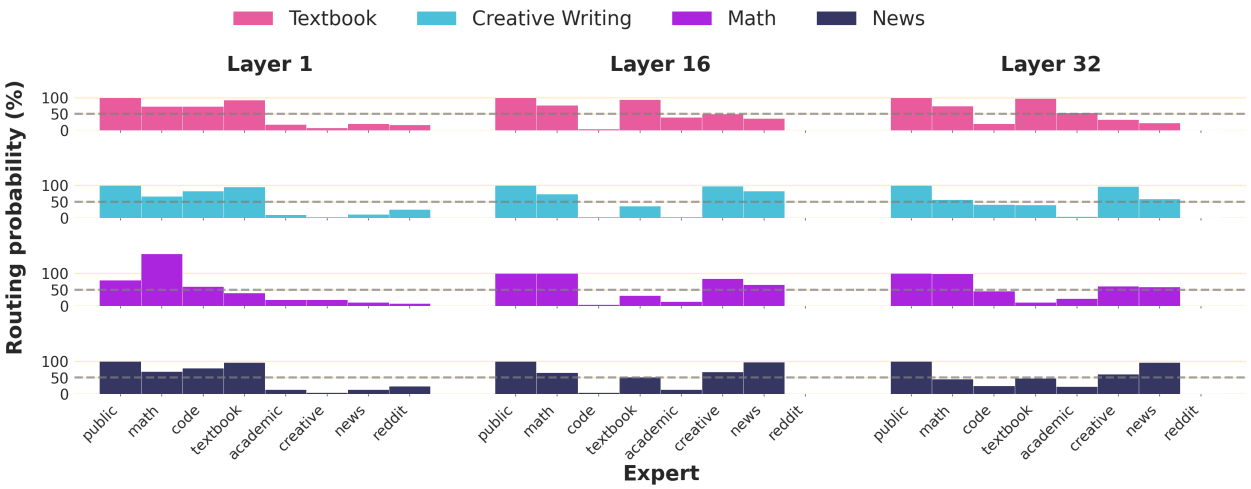

Figure 5.2: Routing pattern analysis. Different domain inputs consistently activate their corresponding expert, while the public expert is frequently co-activated, consistent with the coordinated training design. Different expert combinations are activated at different transformer layers.

Table 5.1: FLEXOLMO outperforms OLMo-2 7B by adding math and code expert modules at equivalent training FLOPs, with especially large gains on math and code tasks.

	Inf. FLOPs	MC9	Gen5	MMLU	MMLU Pro	AGI Eval	BBH	Math	Code	Avg.
Pre-anneal model	1×	74.8	62.8	63.1	32.1	46.8	38.2	17.4	8.7	**43.1**
OLMo-2 7B	1×	77.8	70.2	63.7	31.0	50.4	49.8	42.6	13.3	**49.8**
FLEXOLMO	2.5×	**77.8**	**71.0**	**65.2**	**33.5**	**51.9**	**53.1**	**51.0**	**18.9**	**52.8**

Data extraction risk. A key concern is whether sharing expert weights exposes private training data to extraction attacks [Carlini et al., 2021]. We evaluate empirically on the math expert. The public model (no math data) has a 0.1% extraction rate; the dense math expert alone yields 1.6%; and FLEXOLMO with the math expert included yields 0.7%. Extraction rates are low in all cases. For higher-sensitivity data, applying differential privacy [Dwork and Roth, 2014] during expert training is orthogonal to our architecture, and each data owner can choose independently.

5.1.5 Applications and Impact

FLEXOLMO is being actively applied within the **Cancer AI Alliance**[1] to jointly train a language model on sensitive clinical records across multiple cancer centers, without any patient data leaving the institution of origin. Here, data modularity enables cross-institution scientific collaboration that privacy regulations would

[1] www.canceralliance.ai/blog/caia-federated-learning-cancer-ai

otherwise rule out.

5.2 LLaMaFusion: Efficient Multimodal Model Development

Weijia Shi*, Xiaochuang Han*, Chunting Zhou, Weixin Liang, Xi Victoria Lin, Luke Zettlemoyer, and Lili Yu. 2025. LlamaFusion: Adapting Pretrained Language Models for Multimodal Generation. In *Advances in Neural Information Processing Systems*.

5.2.1 Motivation

The development of multimodal generative models, capable of understanding and generating both text and images, is computationally expensive. State-of-the-art approaches such as Transfusion [Zhou et al., 2024], Chameleon [Team, 2024], and Unified-IO [Lu et al., 2023] train from scratch on joint text-image corpora, requiring enormous compute to achieve proficiency across all modalities. Training a competitive text-only LM already requires training on massive tokens; adding image understanding and generation compounds this cost substantially.

This approach overlooks the massive investment already made in training text-only LMs like Llama-3 [Dubey et al., 2024]. If we can reuse a pretrained text LM as the foundation for a multimodal system, adapting it for image understanding and generation while preserving its text capabilities, we can dramatically reduce the compute required.

The challenge. Naive fine-tuning of a text-only LM on multimodal data causes catastrophic forgetting: the model's text performance degrades substantially as its parameters are updated to handle image tokens. The challenge is to introduce visual capabilities without interfering with existing language capabilities.

5.2.2 The LLaMaFusion Framework

We introduce **LLaMaFusion**, a framework for efficiently developing multimodal generative models by composing a pretrained text LM with dedicated modality-specific image modules.

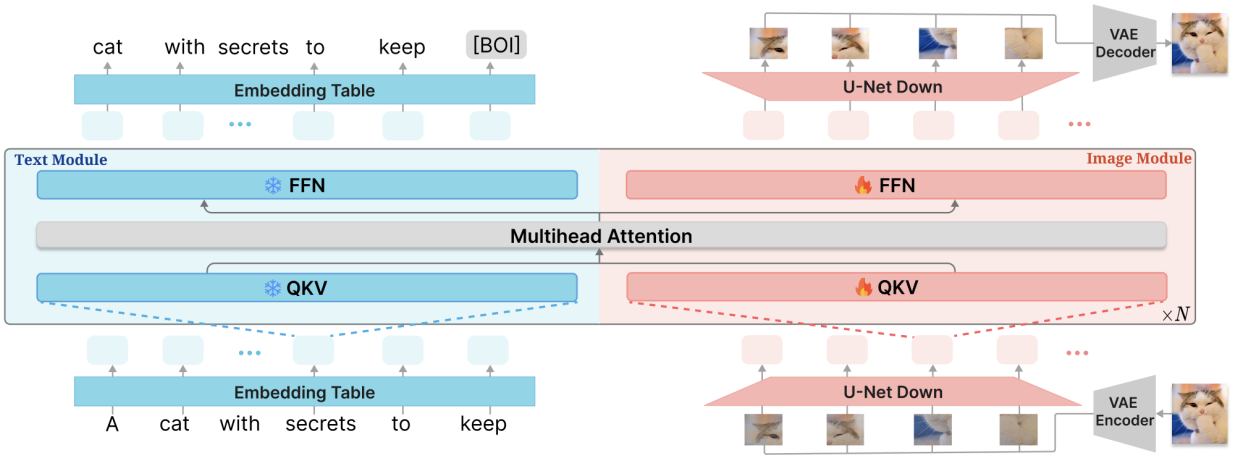

Figure 5.3: LLAMAFUSION architecture. Text and image tokens are processed by modality-specific FFN and QKV layers, while a shared self-attention layer allows cross-modal interaction. All text-specific parameters (original Llama-3 weights) are frozen during training; only the image-specific modules are updated. This complete modality separation preserves language capabilities while enabling visual understanding and generation.

Background: Transfusion. LLAMAFUSION builds on Transfusion [Zhou et al., 2024], which trains a single transformer to jointly handle text via autoregressive language modeling and images via diffusion. For text, the standard LM loss $\mathcal{L}_{\text{LM}}$ is minimized. For images, a raw image is encoded into continuous latent representations $\boldsymbol{x}^{img}$ via a frozen VAE, and the model learns to denoise:

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{\boldsymbol{x}^{img},t,\boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}^{img} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},\, t,\, \boldsymbol{x}^{txt})\|_2^2], \tag{5.3}$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\bar{\alpha}_t$ following a cosine noise schedule. The combined training objective is $\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda \cdot \mathcal{L}_{\text{DDPM}}$. Transfusion trains this objective from scratch, but its text performance lags behind dedicated pretrained LMs. LLAMAFUSION addresses this by reusing a pretrained Llama-3 model.

Architecture: Deep modality separation. The key design is *deep modality separation*: text and image tokens are processed by entirely separate parameters at every layer, preventing image training from corrupting Llama-3's pretrained text representations.

Text tokens $\boldsymbol{x}^{txt}$ and noisy image representations $\boldsymbol{x}_t^{img}$ are projected into hidden states via modality-specific input layers (a text embedding and a U-Net downsampler for images). Separate QKV matrices then project each modality into its own query, key, and value spaces. Cross-modal interaction is enabled by

concatenating keys and values across modalities before computing attention:

$$\boldsymbol{h}_{\mathrm{O}}^{txt} = \mathrm{O}_{\text{text}}\Big(\mathrm{Attn}\left(\boldsymbol{h}_{\mathrm{Q}}^{txt},\ [\boldsymbol{h}_{\mathrm{K}}^{img} \circ \boldsymbol{h}_{\mathrm{K}}^{txt}],\ [\boldsymbol{h}_{\mathrm{V}}^{img} \circ \boldsymbol{h}_{\mathrm{V}}^{txt}]\right)\Big), \tag{5.4}$$

$$\boldsymbol{h}_{\mathrm{O}}^{img} = \mathrm{O}_{\text{img}}\Big(\mathrm{Attn}\left(\boldsymbol{h}_{\mathrm{Q}}^{img},\ [\boldsymbol{h}_{\mathrm{K}}^{txt} \circ \boldsymbol{h}_{\mathrm{K}}^{img}],\ [\boldsymbol{h}_{\mathrm{V}}^{txt} \circ \boldsymbol{h}_{\mathrm{V}}^{img}]\right)\Big), \tag{5.5}$$

where $\circ$ denotes concatenation and a hybrid attention mask applies causal masking to text positions and bidirectional masking to image positions. Modality-specific FFNs and layer normalizations process each modality's hidden states independently after attention. Text hidden states are projected to next-token logits via Llama-3's LM head; image hidden states are projected to predicted noise via a U-Net upsampler.

Training: Decoupled learning rates. All text-specific parameters (Llama-3's embedding, QKV, FFN, and LM-head weights) are **frozen** throughout training ($\eta_{\text{text}} = 0$), while only image-specific parameters are updated ($\eta_{\text{img}} > 0$). This decoupled learning rate scheme guarantees that language capabilities are fully preserved: text parameters never receive gradients from the image objective, so visual training cannot cause catastrophic forgetting. Because text parameters are frozen, there is also no need to include text-only data in the training mix, substantially reducing compute cost.

5.2.3 Results

We compare LLAMAFUSION to Transfusion [Zhou et al., 2024] trained from scratch on the same image data, initializing LLAMAFUSION from Llama-3-8B [Dubey et al., 2024].

Image understanding. LLAMAFUSION achieves **20% better image understanding** (as measured by VQA accuracy on standard benchmarks) compared to a from-scratch Transfusion model trained with the same compute budget.

Image generation. LLAMAFUSION achieves **3.6% better image generation quality** (as measured by FID and CIDEr on COCO) compared to Transfusion at comparable compute.

Compute efficiency. LLAMAFUSION achieves these improvements using only **50% of the FLOPs** of the comparable Transfusion baseline, because the text modules are frozen and the model does not need to be

Table 5.2: LLAMAFUSION preserves Llama-3's text performance while adding strong multimodal capabilities. At 0.5× image FLOPs, LLAMAFUSION matches or exceeds Transfusion (1× FLOPs) on all tasks.

		Language-only Evaluation			Image Understanding	Image Generation (w/o | w CFG)	
Model	Rel. FLOPs	HellaSwag↑	SIQA↑	WinoGrande↑	CIDEr↑	FID↓	CLIP↑
Llama-3		60.0	48.1	72.8			
Transfusion	1×	51.0	42.3	64.3	32.0	14.4 | 8.70	22.1 | 24.4
LLAMAFUSION	0.5×	60.0	48.1	72.8	38.3	13.9 | 8.75	22.0 | 24.4
LLAMAFUSION	1×	60.0	48.1	72.8	**38.4**	**14.0 | 8.61**	**22.1 | 24.4**

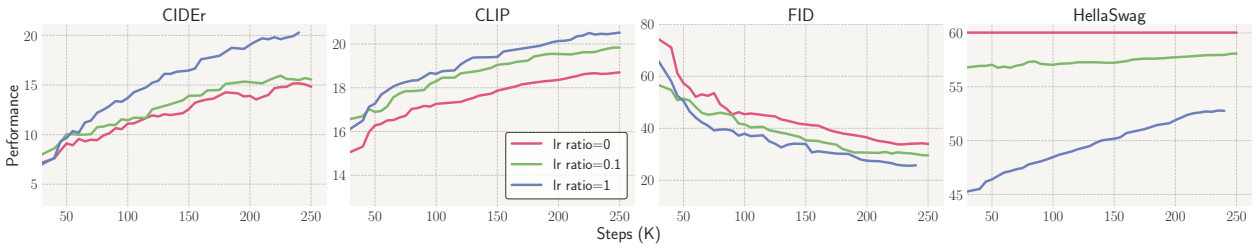

Figure 5.4: Naive Llama-3 fine-tuning (no modality separation) with varying learning rate ratio $\frac{\eta_{\text{text}}}{\eta_{\text{img}}}$. Equal learning rates (ratio $= 1$) improve image performance but cause a persistent 7% drop in text performance. Reducing the text learning rate (ratio < 1) partially mitigates text degradation but reduces image capabilities. No setting satisfies both objectives simultaneously.

trained on text-only data.

Text performance. LLAMAFUSION preserves Llama-3's text-only performance exactly, outperforming Transfusion by 11.6% on text benchmarks. Because the text parameters never receive gradients from the image objective, visual training cannot degrade language capabilities.

5.2.4 Analysis: Why Does Modality Separation Work?

We conduct ablation studies comparing three architectural designs, all initialized from Llama-3-8B: (1) **no separation**, a single dense model with shared QKV, FFN, and O weights; (2) **shallow separation**, modality-specific FFNs only, with shared QKV and O; and (3) **deep separation**, modality-specific FFNs and QKV projections (LLAMAFUSION). For each, we vary the learning rate ratio $\frac{\eta_{\text{text}}}{\eta_{\text{img}}} \in \{0, 0.1, 1\}$.

Figure 5.4 shows that naive fine-tuning of the dense Llama-3 model faces an unavoidable trade-off: using the same learning rate for text and image components continuously improves image performance but causes a

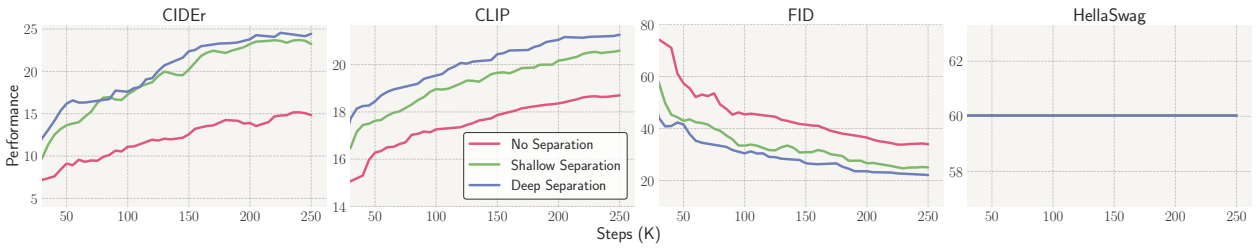

Figure 5.5: No separation vs. shallow separation vs. deep separation with frozen text modules ($\frac{\eta_{\text{text}}}{\eta_{\text{img}}} = 0$). Deep modality separation outperforms shallow separation on image generation, and both substantially outperform the dense model. Since text modules are frozen, all models preserve Llama-3's original language capabilities.

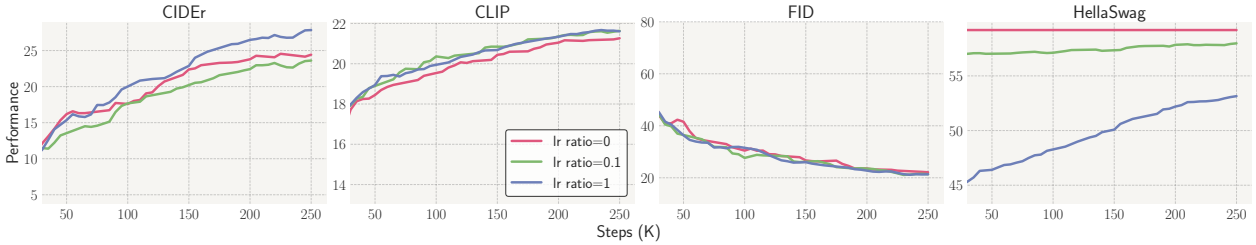

Figure 5.6: Deep modality separation with varying learning rate ratio $\frac{\eta_{\text{text}}}{\eta_{\text{img}}}$. With deep separation, freezing the text modules ($\frac{\eta_{\text{text}}}{\eta_{\text{img}}} = 0$) preserves language capabilities *and* achieves the best image performance, the opposite of what happens in the dense model (Figure 5.4).

persistent 7% drop in HellaSwag. Reducing the text learning rate recovers some language performance but degrades image learning.

Figure 5.5 shows that once text modules are frozen, both separation approaches substantially outperform the dense model on image benchmarks, with deep separation (modality-specific FFNs and QKV) outperforming shallow separation especially on image generation. The QKV projections determine how text and image representations are mixed in attention; sharing them causes image training to corrupt the text-adapted representation spaces.

Figure 5.6 reveals the key insight: with deep modality separation, freezing the text modules ($\frac{\eta_{\text{text}}}{\eta_{\text{img}}} = 0$) is not just safe — it is the *optimal* configuration, achieving the best image performance while guaranteeing perfect text preservation. This is the opposite of what happens in the dense model, where freezing prevents the model from learning image representations effectively.

5.3 Related Work

Mixture-of-experts language models. Sparse mixture-of-experts architectures [Shazeer et al., 2017; Fedus et al., 2022; Lepikhin et al., 2021] replace dense FFN layers with collections of expert FFNs, routing each token to a subset of experts via a learned router. These models achieve better performance per FLOP than dense models by allowing parameters to specialize to different input patterns. However, standard MoE training is fully synchronized: all experts and the router are trained jointly, requiring communication across all GPUs. FLEXOLMO achieves the benefits of expert specialization through fully asynchronous training, at the cost of requiring a shared anchor rather than a joint router.

Embarrassingly parallel and distributed training. Branch-Train-Merge [Li et al., 2022] first demonstrated that separately training expert LMs on different domains and merging their parameters can match or exceed jointly trained models. FLEXOLMO extends this to the private-data setting where different organizations train experts independently. Federated learning [Konečnỳ et al., 2016] similarly trains models without centralizing data, but requires synchronized gradient communication across participants; FLEXOLMO eliminates this requirement.

Model merging. Model soup [Wortsman et al., 2022] showed that averaging the weights of multiple fine-tuned models can improve robustness over any individual model. Task vectors [Ilharco et al., 2023] represent fine-tuning as a direction in weight space, enabling task composition via vector arithmetic. Fisher-weighted averaging [Matena and Raffel, 2022] weights parameter contributions by their Fisher information, prioritizing parameters most important to each model's task. TIES-Merging [Yadav et al., 2023] addresses interference between models by resolving sign conflicts and trimming redundant parameters before merging. MaTS [Tam et al., 2024] identifies task-specific parameter subspaces and matches models within those subspaces, yielding improved multi-task performance. FLEXOLMO builds on these ideas but operates in the more challenging setting where experts are trained on completely disjoint, private datasets, requiring the shared anchor algorithm to ensure experts remain compatible despite never seeing each other's data.

Unified models for multimodal generation. Unified multimodal models differ primarily in how they represent images. One line of work uses vector-quantized discrete tokens [Van Den Oord et al., 2017;

Esser et al., 2021]: Chameleon [Team, 2024], Unified-IO 1&2 [Lu et al., 2023, 2024a], and Show-O [Xie et al., 2024] all process interleaved text and image tokens autoregressively. Transfusion [Zhou et al., 2024] instead combines autoregressive text generation with continuous diffusion for images in a single model. LLAMAFUSION builds on Transfusion but avoids training from scratch by adapting a pretrained Llama-3 model.

Modality-specific processing. To reduce interference between text and image modalities, several works introduce modality-specific components. VLMo [Wang et al., 2021] and BEiT-3 [Wang et al., 2022] replace shared FFNs with modality-specific experts while keeping attention shared. Concurrent to LLAMAFUSION, Mixture-of-Transformers [Liang et al., 2024] independently shows that separating both FFNs and attention, the same design choice as LLAMAFUSION, is effective for large-scale multimodal pretraining and image generation.

5.4 Conclusion

This chapter focuses on the decentralized method for language model development: rather than training a monolithic model on centralized data, we build models from independently trained, composable components. FLEXOLMO demonstrates that independently trained expert modules, merged without any joint training, improve over the public baseline, outperforming model soup and ensembling. Because components are trained independently, they can be added, updated, or removed without touching the rest of the model, providing data control that machine unlearning (Chapter 2) cannot reliably deliver. Additionally, LLAMAFUSION applies data modularity to multimodal development, showing that freezing a pretrained text LM and training only image-specific modules achieves better image understanding and better image generation than training from scratch, at 50% of the compute cost.

Chapter 6

Conclusion

Language models have become a cornerstone of modern AI systems. However, the monolithic paradigm, training a single large model indiscriminately on all available data, carries structural limitations that scale alone cannot resolve. When all knowledge is fused into a single set of parameters, the resulting systems are factually unreliable, opaque about what they have memorized, and unable to protect the privacy of their training data. This thesis presents a research shift centered on *modularity* as a principled response to these limitations.

In Chapter 2, we characterize the limits of the monolithic paradigm, focusing on training data memorization and its consequences. We introduce a reference-free membership inference method for pretraining data detection that requires only black-box token probabilities [Shi et al., 2024a]. Applied to GPT-3, it reveals strong evidence of copyrighted content memorization and benchmark contamination. We then introduce a six-way evaluation framework for machine unlearning, showing that current algorithms fail to meet basic expectations around privacy leakage, utility, and scalability [Shi et al., 2025c]. These findings motivate the architectural alternatives developed in the subsequent chapters.

This dissertation presents a new framework that builds language models from modular, independently trained components, and moves away from monolithic treatment of training data. The lack of reliable data control in monolithic LMs motivates the techniques we present in Chapter 3, where we develop model modularity through retrieval augmentation. REPLUG introduces the framework for retrieval-augmented generation with black-box LMs [Shi et al., 2024d]. We then improve how LMs use retrieved content through

a pretraining method that trains on topically related documents [Shi et al., 2024c] and a context-aware decoding algorithm that encourages faithful attention to context [Shi et al., 2024b]. We additionally contribute INSTRUCTOR, a single embedding model that adapts to any task through natural language instructions [Su et al., 2023], and RECOMP, which compresses retrieved passages and selectively skips retrieval to reduce inference cost [Xu et al., 2024].

These retrieval-augmented techniques lead naturally into Chapter 4, where we extend model modularity to the visual domain. Retrieval-augmented multimodal generation shows that retrieved image-text references can ground visual outputs more faithfully and efficiently than parametric knowledge alone [Yasunaga et al., 2023]. We further introduce *visual chain-of-thought* through Visual Sketchpad, which enables LMs to dynamically invoke specialist vision models and use their outputs as intermediate reasoning steps [Hu et al., 2024]. Applied to GPT-4o, SKETCHPAD yields significant accuracy improvements on mathematical and visual reasoning benchmarks.

In Chapter 5, we challenge the assumption that all model parameters must be trained on centralized data. FLEXOLMO introduces a mixture-of-experts architecture where data owners independently *train* expert modules anchored to a shared public model, and *merge* them without any joint training or data sharing [Shi et al., 2025a]. The merged model substantially improves over the public baseline across a diverse set of tasks. LLAMAFUSION applies data modularity to multimodal development: composing a frozen pretrained Llama-3 model with independently trained image modules achieves better image understanding and image generation than training from scratch, at a fraction of the compute cost [Shi et al., 2025b]. Overall, our results suggest that treating data modularly is valuable beyond privacy alone: it also enables more efficient scaling and stronger guarantees on model behavior.

Looking forward, the augmented and modular paradigm developed in this thesis opens up several directions for building the next generation of language models. We highlight three.

Designing socially responsible language models. The modular paradigm provides a new foundation for socially responsible language models that address critical challenges in privacy, safety, and copyright. FLEXOLMO, which enables fine-grained data use based on legal constraints, is a first step. Future work can intervene at both training and inference time to ensure models use data responsibly. At training time, an important challenge is ensuring that training data is not memorized or extractable; applying differentially

private training algorithms [Dwork and Roth, 2014] is a promising direction, but understanding their interplay with modular architectures remains an open problem. At inference time, dynamic and fine-grained generation controls, such as controllable decoding methods that steer outputs in real time to align with complex legal and safety policies, could enable source attribution for copyright, prevent leakage of private data, and support safe personalization.

A science of scaling for modular models. This new class of modular models will demand a new science of scaling across both training and inference, distinct from the established scaling laws for monolithic models [Kaplan et al., 2020; Hoffmann et al., 2022]. At training time, future work can derive scaling laws that connect data, parameters, and module composition to downstream performance. At inference time, modularity can be combined with test-time scaling, building on recent results showing how much models benefit from additional inference-time compute [Muennighoff et al., 2025]. A key question is how to distribute inference budgets across components, such as retrieval depth, tool calls, and planning steps, to reach the quality-latency-cost frontier. Reliable scaling laws would enable end-to-end co-optimization of all modules, giving practitioners a principled guide for precisely what to scale, by how much, and in what order.

Interactive multimodal learning for embodied AI. We envision future LMs learning not just from passive observation, but through active interaction with the visual world, much as humans do. Building on our work on unified multimodal understanding and generation [Shi et al., 2025b] and visual reasoning [Hu et al., 2024], future research can develop methods that learn directly from rich multimodal feedback, such as visual state changes and action outcomes, allowing models to build a grounded understanding of the world and to predict or even generate its next state in both text and images. The goal is to create models that can perceive, reason about, and act within the world, opening the way to advances in robotics, embodied AI, and human-computer interaction.

Bibliography

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. 2022. CM3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*.

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. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large scale autoregressive language modeling with mesh-tensorflow. *Zenodo*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*.

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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *USENIX Security Symposium*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. 2023a. Re-imagen: Retrieval-augmented text-to-image generator. In *International Conference on Learning Representations*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2023b. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Abhimanyu Dubey et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in LLMs. *arXiv preprint arXiv:2310.02238*.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(1):5232–5270.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Yun Ma, and Ranjay Krishna. 2024. BLINK: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace

He, Anish Thite, Noa Nabeshima, et al. 2021. The pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. 2024. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems*.

Hugging Face. 2023. Open LLM leaderboard. https://huggingface.co/spaces/open-llm-leaderboard.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *International Conference on Learning Representations*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *European Chapter of the Association for Computational Linguistics*.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2023. Knowledge unlearning for mitigating privacy risks in language models. In *Annual Meeting of the Association for Computational Linguistics*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Lam. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*.

Peter Kairouz, H. Brendan McMahan, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Empirical Methods in Natural Language Processing*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization. In *Annual Meeting of the Association for Computational Linguistics*.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. 2024. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Scott Yih. 2024. RA-DIT: Retrieval-augmented dual instruction tuning. In *International Conference on Learning Representations*.

Alisa Liu and Jiacheng Liu. 2023. The MemoTrap dataset. https://github.com/inverse-scaling/prize/blob/main/data-release/README.md.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*.

Ziyu Liu, Yuhang Zang, Yushan Zou, Zijian Liang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. 2025. Visual agentic reinforcement fine-tuning. *arXiv preprint arXiv:2505.14246*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, Sameer Singh, Tatsunori Hashimoto, Siddharth Mishra, and Arman Cohan. 2021. Entity-based knowledge conflicts in question answering. In *Empirical Methods in Natural Language Processing*.

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.

Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. 2023. Unified-IO: A unified model for vision, language, and multi-modal tasks. In *International Conference on Learning Representations*.

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 *International Conference on Learning Representations*.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Annual Meeting of the Association for Computational Linguistics*.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. 2024. TOFU: A task of fictitious unlearning for LLMs. *arXiv preprint arXiv:2401.06121*.

Michael S. Matena and Colin A. Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, pages 17703–17716.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. In *Empirical Methods in Natural Language Processing*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Empirical Methods in Natural Language Processing*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Sycophancy to subterfuge: Investigating reward tampering in language models. In *Advances in Neural Information Processing Systems*.

Alec Radford, Jong Woon 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*.

Colin Raffel. 2023. Building machine learning models like open source software. *Communications of the ACM*, 66(2):38–40.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024a. Detecting pretraining data from large language models. In *International Conference on Learning Representations*.

Weijia Shi, Akshita Bhagia, Kevin Farhat, Niklas Muennighoff, Pete Walsh, Jacob Morrison, Dustin Schwenk, Shayne Longpre, Jake Poznanski, Allyson Ettinger, et al. 2025a. FlexOlmo: Open language models for flexible data use. In *Advances in Neural Information Processing Systems*. Spotlight.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024b. Trusting your evidence: Hallucinate less with context-aware decoding. In *North American Chapter of the Association for Computational Linguistics*.

Weijia Shi, Xiaochuang Han, Chunting Zhou, Weixin Liang, Xi Victoria Lin, Luke Zettlemoyer, and Lili Yu. 2025b. LMFusion: Adapting pretrained language models for multimodal generation. In *Advances in Neural Information Processing Systems*.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. 2025c. MUSE: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations*.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Wen-tau Yih, and Mike Lewis. 2024c. In-context pretraining: Language modeling beyond document boundaries. In *International Conference on Learning Representations*. Spotlight.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024d. REPLUG: Retrieval-augmented black-box language models. In *North American Chapter of the Association for Computational Linguistics*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, pages 3–18.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. ViperGPT: Visual inference via python execution for reasoning. In *IEEE/CVF International Conference on Computer Vision*.

Derek Tam, Mohit Bansal, and Colin Raffel. 2024. Merging by matching models in task parameter subspaces. *Transactions on Machine Learning Research*.

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

Gemini Team. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, et al. 2022. Image as a foreign language: BEiT pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*.

Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. 2021. VLMo: Unified vision-language pre-training with mixture-of-modality-experts. *arXiv preprint arXiv:2111.02358*.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

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

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard Hua Yang, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. 2019. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carlin, Simon Kornblith, et al. 2022. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. 2024. Show-o: One single transformer to unify multimodal understanding and generation.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: Improving retrieval-augmented LMs with compression and selective augmentation. In *International Conference on Learning Representations*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. 2023. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, pages 7093–7115.

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*.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022a. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xiaojin Zhang, Yan Kang, Kai Chen, Lixin Fan, and Qiang Yang. 2022b. Trading off privacy, utility, and efficiency in federated learning. *ACM Transactions on Intelligent Systems and Technology*, 14:1–32.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. 2024. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556. Association for Computational Linguistics.

Appendix A

Additional Results for Pretraining Data Detection

This appendix provides additional experimental results and analysis for the MIN-K% PROB method described in Chapter 2 (Section 2.1).

A.1 Additional Models and Datasets

We evaluate MIN-K% PROB across a wider range of LMs, including OPT [Zhang et al., 2022a], GPT-Neo [Black et al., 2021], and the full LLaMA-1 [Touvron et al., 2023] model family. Table A.1 reports AUC on WIKIMIA (original setting) for each model. Results are consistent with those reported in the main chapter: MIN-K% PROB consistently outperforms the strongest reference-free baseline (average log-likelihood) across all model families and sizes, and performance improves monotonically as model size increases, confirming that larger models memorize more aggressively.

A.2 Sensitivity to the k Parameter

The k parameter in MIN-K% PROB controls the fraction of minimum-probability tokens used. We evaluate performance across $k \in \{5, 10, 15, 20, 30\}$. Table A.2 reports AUC for three representative models. We find that $k = 20$ generally provides the best performance, with robustness to modest changes in k: performance

Table A.1: AUC on WIKIMIA (original setting) for MIN-K% PROB versus the average log-likelihood baseline (LOSS) across the OPT, GPT-Neo, and LLaMA-1 families. **Bold** = best per row. Detection performance improves consistently with model size.

Family	Model	LOSS	MIN-K% PROB
OPT	1.3B	0.61	**0.64**
	2.7B	0.62	**0.66**
	6.7B	0.64	**0.68**
	66B	0.66	**0.71**
GPT-Neo	1.3B	0.62	**0.65**
	2.7B	0.63	**0.67**
	20B	0.70	**0.76**
LLaMA-1	7B	0.66	**0.70**
	13B	0.68	**0.71**
	30B	0.70	**0.74**
	65B	0.71	**0.74**

Table A.2: AUC on WIKIMIA (original setting) as a function of the MIN-K% PROB hyperparameter k (% of lowest-probability tokens). **Bold** = best per row. $k = 20$ is consistently near-optimal.

Model	$k = 5$	$k = 10$	$k = 15$	$k = 20$	$k = 30$
Pythia-2.8B	0.64	0.65	0.66	**0.67**	0.66
NeoX-20B	0.72	0.74	0.75	**0.76**	0.75
LLaMA-65B	0.71	0.72	0.73	**0.74**	0.73

is stable within roughly ± 0.01 AUC for $k \in \{15, 20, 30\}$. The optimal k varies slightly across models and domains.

A.3 Text Length Analysis

We analyze how detection performance varies with the length of the evaluated text. Longer texts provide more tokens from which to compute the minimum-probability statistics, leading to more reliable detection. Table A.3 reports AUC across text-length buckets for LLaMA-65B. Detection performance is substantially lower for very short texts (fewer than 50 tokens) and stabilizes for texts of 200 or more tokens.

Table A.3: AUC on WIKIMIA for LLaMA-65B as a function of text length (number of tokens). Detection is unreliable for very short texts and stabilizes beyond 200 tokens.

Length (tokens)	< 50	50–100	100–200	≥ 200
LOSS	0.58	0.64	0.69	0.71
MIN-K% PROB	0.61	0.69	0.73	0.76

Appendix B

Additional Results for Machine Unlearning Evaluation

This appendix provides additional results from the MUSE framework described in Chapter 2 (Section 2.2).

B.1 Full Results Across All Eight Algorithms

We report the full six-dimensional evaluation profile for all eight unlearning algorithms on both the BOOKS and NEWS unlearning scenarios. Table B.1 extends Table 2.2 in the main chapter with the two deployer-side criteria omitted there: C5 (scalability, measured as the relative utility drop when the forget set is scaled from a single passage to an entire book/corpus) and C6 (sustainability, measured as the relative utility drop after four sequential unlearning requests). The full results confirm the findings reported in the main chapter: privacy leakage (C3) is near-universal across algorithms, the utility-forgetting trade-off is stark, and both scalability and sustainability degrade sharply for the most aggressive forgetting methods.

B.2 Interaction Between Forget Set and Retain Set

One of the novel aspects of the MUSE NEWS scenario is that the forget set (BBC news articles) and the retain set (other BBC articles) have significant topical overlap. We analyze how this overlap affects the six evaluation criteria. Table B.2 reports KnowMem on the forget set ($\mathcal{D}_u$, lower is better) as a function of the

Table B.1: Full six-criterion MUSE profile for all eight algorithms on BOOKS and NEWS. C1 = VerbMem on $\mathcal{D}_u$ (↓), C2 = KnowMem on $\mathcal{D}_u$ (↓), C3 = PrivLeak (best $\in [-5\%, 5\%]$), C4 = KnowMem on $\mathcal{D}_r$ (↑), C5 = utility drop at scale (↓), C6 = utility drop after 4 sequential requests (↓). Only Retrain satisfies C3.

	BOOKS						NEWS					
Method	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
Target f_t	99.8	59.4	−57.5	69.0	–	–	58.4	63.9	−99.8	55.2	–	–
Retrain f_r	14.3	28.9	0.0	68.9	–	–	20.8	33.1	0.0	55.0	–	–
GA	0.0	0.0	−26.6	0.0	41.2	58.3	0.0	0.0	5.2	0.0	38.7	55.1
GA_{GDR}	4.1	26.6	−26.5	23.3	33.9	47.6	4.9	31.0	108.1	27.3	31.2	44.8
GA_{KLR}	22.7	47.5	−27.3	41.4	30.5	41.0	27.4	50.2	−96.1	44.8	28.4	39.2
NPO	0.0	0.0	5.1	0.0	39.4	55.8	0.0	0.0	4.8	0.0	36.9	52.7
NPO_{GDR}	0.0	19.4	−3.9	24.9	32.1	45.2	1.2	23.1	−2.5	28.7	30.0	42.9
NPO_{KLR}	19.8	41.3	−22.6	39.7	29.8	40.3	24.1	44.6	−89.4	43.2	27.9	38.6
Task Vector	8.2	31.7	−19.2	30.1	35.7	50.1	11.3	36.5	−71.3	33.9	33.4	47.0
WHP	25.4	49.1	−31.8	44.2	27.6	37.5	29.8	52.7	−93.7	47.1	26.1	35.9

Table B.2: Effect of forget/retain topical overlap on residual forget-set knowledge under MUSE NEWS, using NPO_{GDR}. KnowMem on $\mathcal{D}_u$ (↓, lower means more successful forgetting) rises as overlap increases, while retain-set utility (KnowMem on $\mathcal{D}_r$) is largely preserved.

Forget/Retain Overlap	Low	Medium	High
KnowMem on $\mathcal{D}_u$ (↓)	11.4	18.9	26.8
KnowMem on $\mathcal{D}_r$ (↑)	27.9	28.4	28.7

topical overlap between forget and retain sets, holding the unlearning algorithm fixed (NPO_{GDR}). We find that knowledge memorization is harder to fully remove when the retain set covers similar topics: residual forget-set knowledge rises from 11.4 at low overlap to 26.8 at high overlap. The model retains knowledge partly because it is present in the retain set, not solely from the forget set, exposing a fundamental tension between forgetting and retain-set utility preservation.

Appendix C

Additional Results for REPLUG

This appendix provides additional analysis for REPLUG, described in Chapter 3 (Section 3.1).

C.1 Analysis of Retrieved Documents

We analyze the types of documents retrieved by Contriever and by the LSR-trained retriever for a sample of queries. Table C.1 reports, for 500 sampled Pile queries, the fraction of retrieved documents that fall into each relevance category, as judged by whether the document contains information directly useful for predicting the query continuation. The LSR-trained retriever learns to prioritize documents that provide information directly relevant to predicting the continuation of the query (62.4% directly relevant vs. 41.8% for Contriever), even when these documents are not the most semantically similar in embedding space.

C.2 Computational Cost Analysis

We analyze the inference cost of REPLUG as a function of the number of retrieved documents k. Because each retrieved document is processed in parallel, the latency scales sublinearly with k on multi-GPU setups. Table C.2 reports relative latency (normalized to the no-retrieval baseline) and bits-per-byte improvement on the Pile for GPT-3 Curie as k varies. Most of the quality gain is realized by $k = 10$, after which returns diminish while latency continues to grow.

Table C.1: Distribution of retrieved-document relevance categories (% of top-1 retrievals over 500 sampled Pile queries) for Contriever versus the REPLUG LSR-trained retriever. LSR training shifts retrievals toward documents that are directly useful for next-token prediction.

Retrieved-document category	**Contriever**	**LSR retriever**
Directly relevant (useful for continuation)	41.8	**62.4**
Topically similar but not useful	38.5	26.1
Loosely related	14.2	8.7
Irrelevant	5.5	2.8

Table C.2: Inference cost versus quality for REPLUG (GPT-3 Curie) as the number of retrieved documents k increases. Latency is normalized to the $k = 0$ (no-retrieval) baseline; documents are encoded in parallel across GPUs, so latency grows sublinearly. BPB on the Pile (↓).

# Retrieved docs k	0	1	5	10	20
Relative latency (×)	1.00	1.18	1.43	1.71	2.34
BPB on the Pile (↓)	0.88	0.86	0.85	0.85	0.84

Appendix D

Additional Results for Visual Sketchpad

This appendix provides additional analysis for SKETCHPAD, described in Chapter 4 (Section 4.1).

D.1 Tool Usage Analysis

We analyze which tools are most commonly invoked by GPT-4o with SKETCHPAD across different task categories. Table D.1 reports the fraction of solved instances in which each tool is invoked, broken down by task category. For geometry tasks, bounding-box drawing and line/auxiliary-annotation tools are most frequent; for depth-estimation tasks, depth-map generation is the primary tool; and for counting tasks, detection models are most useful.

D.2 Failure Cases

We analyze cases where SKETCHPAD does not improve or hurts performance. Table D.2 reports the distribution of failure modes over 150 manually inspected error cases. Common failure modes include: (1) invoking tools that return irrelevant information for the specific task, (2) misinterpreting tool outputs, and (3) over-relying on visual artifacts when text-based reasoning would be more direct.

Table D.1: Tool-invocation frequency (% of solved instances in which the tool is used) for GPT-4o + SKETCHPAD, broken down by task category. Each instance may invoke multiple tools, so rows need not sum to 100%.

Tool	Geometry	Depth	Counting	Spatial
Bounding-box / line drawing	78.4	9.2	31.7	44.6
Object detection	12.1	6.5	86.3	52.8
Depth-map generation	3.0	91.7	4.1	28.4
Segmentation	6.5	14.3	22.9	19.7

Table D.2: Distribution of SKETCHPAD failure modes over 150 manually inspected error cases (GPT-4o + SKETCHPAD).

Failure mode	% of errors
Irrelevant tool invocation	34.7
Misinterpreting tool output	28.0
Over-reliance on visual artifacts	20.7
Tool execution / API error	10.0
Correct tools, flawed final reasoning	6.6