Statistically principled measurement of large language models by spiking the training data

by

Johnny Tian-Zheng Wei

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

May 2026

To the City of Angels and its most scenic overlook, on which I left a piece of my soul.

Acknowledgments

A good place to start is gratitude. I would like to first thank my advisor, Robin Jia. Robin took me on as his first student, and together we have seen success, failure, and the transformation of our field several times over. I hope to speak for both of us when I say that none of it was easy, but my prevailing sense is that my PhD experience was true and a journey not short of intellectual and personal growth. Robin always made space for me to explore and challenged me to go beyond my own expectations; for that I am grateful.

I owe much to my collaborators, who not only contribute to my work but also inspire me. I would like to thank my legal collaborators Frederike Zufall and Jonathan Choi. I never received a formal education in law, but both Frederike and Jon gave me countless oral lessons in legal thinking. With Frederike I learned several difficult lessons, including on collaboration and interdisciplinary work, and I want to emphasize my gratitude. On several projects, Ameya Godbole was my most formidable collaborator. Ameya inspires me to be more meticulous and responsible through his example. I thank all the collaborators who made my work possible, including Mohammed Aflah Khan, Xiaoyuan Zhu, James Flemings, Willie Neiswanger, Krishna Gummadi, Xinyue Cui, Swabha Swayamdipta, and Tom Kocmi.

I would like to reserve a special thanks for my junior collaborators. My students Ryan Wang, Maggie Wang, and Jerry Li were each co-first authors on major parts of this thesis. I will never forget the final hour submissions with each of you, although I must apologize for keeping you up so late. While you were learning from me, I was learning as well, and my students often rose to the occasion to complement me in my weaker areas. I was at my most inexperienced when mentoring Ryan, and for all that I learned, our time together was special. Several more undergraduates contributed to my work, including Nitya Kashyap and Xiaoyuan Zhu; I am grateful for all the steps we took together.

On many occasions my experience with the academic community was positive. Though I

see the flaws, I don't take our culture of kindness, openness, and service for granted. I want to thank the communities of *CL, FAccT, SoCalNLP, and that of any other conference I had the privilege to attend. One of the biggest challenges in my PhD was overcoming insecurity. Finding confidence was a gradual process and shaped through the many positive interactions I had with other researchers. I want to extend my sincere gratitude to anyone who has taken the time to speak with me or engage with my research. I also want to thank my fellow workshop organizers Eric Wallace, Yangsibo Huang, Tiago Pimentel, Pratyush Maini, Verna Dankers, and Pietro Lesci. When asked about the workshop, I can't help but describe it as the perfect academic day against the backdrop of the beautiful Vienna summer.

Many more supported my academic journey that do not fall into these categories. I see you and I am grateful. Yan Liu served as a committee member for my qualification, proposal, and defense. Nate Fast served as a committee member for my defense. Yanai Elazar, Qinyuan Ye, Ting-yun Chang, and Gustavo Carvalho provided feedback for my work on multiple occasions. For those with whom collaboration did not go as we had hoped, please know that I also wish you well. This includes my former advisor Nanyun Peng; thank you for giving me the chance to pursue a PhD at USC in the first place.

I know I will look back on my PhD as a beautiful time in my life. I was blessed with the constant support of friends and family. I often commuted to USC campus in the radiant LA weather. LA showed me new facets of being. It was the right place to undertake this substantial work, and I am grateful for all that had to come together to make it so.

Table of Contents

Acknowledgments ii

List of Tables vi

List of Figures x

Abstract xiii

Chapter 1:
Introduction 1
1.1 Model metrology 1
1.2 LLM memorization 2

Chapter 2:
The case for spiking 3
2.1 A statistical perspective 3
2.1.1 Hypothesis testing 3
2.1.2 Causal regressions 4
2.1.3 Case study in membership inference 5
2.2 A legal perspective on spiking 7
2.2.1 Behavioral and structural perspectives 8
2.2.2 Case study on Pythia 9
2.2.3 Spiking as a structural intervention 12

Chapter 3:
The Hubble model suite 14
3.1 Introduction 14
3.2 Perturbation Design across Risk Domains 16
3.2.1 Copyright 16
3.2.2 Privacy 18
3.2.3 Test set contamination 19
3.3 The Hubble Suite 20
3.3.1 Pretraining Data 20
3.3.2 Models 22
3.3.3 Evaluations 23
3.4 Domain-agnostic Results 25

3.5 Use Cases of Hubble . . . 27
3.5.1 Hubble as an MIA Benchmark . . . 28
3.5.2 HUBBLE as an Unlearning Benchmark . . . 29
3.6 Conclusion . . . 31

Chapter 4:
Correcting test set contamination . . . 34
4.1 Introduction . . . 34
4.2 Related work . . . 36
4.3 Simulating contamination . . . 37
4.3.1 Estimators and predictors . . . 37
4.3.2 Data generation process . . . 39
4.3.3 Phase diagrams . . . 41
4.4 Benchmarking predictors . . . 43
4.4.1 Memorization predictors . . . 43
4.4.2 Correctness predictors . . . 45
4.4.3 Correcting estimates with predictors . . . 46
4.5 Practical considerations . . . 48
4.6 Conclusion . . . 50

Chapter 5:
Conclusion . . . 51
5.1 Future directions . . . 51
5.2 Limitations . . . 52
5.3 Governance . . . 53

Appendix A:
Supplementary material for Chapter 3 . . . 55
A.1 Perturbations . . . 55
A.1.1 List of Datasets . . . 55
A.1.2 Perturbation Statistics . . . 58
A.1.3 Decontamination . . . 59
A.2 Training . . . 60
A.2.1 Model Architecture . . . 60
A.2.2 Setup . . . 60
A.2.3 GPU Hours . . . 62
A.3 General Evaluation . . . 62
A.4 Additional HUBBLEMIA Results . . . 66
A.5 Additional HUBBLEUNLEARNING results . . . 68

Appendix B:
Supplementary material for Chapter 4 . . . 71
B.1 Non-spiking Baseline: EPG . . . 71

Bibliography . . . 74

List of Tables

3.1 **HUBBLE perturbation datasets on Hugging Face, grouped by domain and data type.** Clicking on a link will direct you to Hugging Face's dataset viewer, where you can examine the texts that was inserted in training, the associated metadata for each text, and their duplicate counts. 17

3.2 **ROC AUC scores of baseline MIAs for our largest perturbed model (8B, 500B tokens).** *Dup* indicates the duplication level of members. *Dup* $\neq$ *0* treats all inserted perturbations as members. Non-members are always drawn from perturbations inserted 0 times. As duplication increases, memorization becomes stronger, and it becomes easier for membership inference attacks (MIA) to distinguish between members and non-members. See Appendix A.4 for more HUBBLEMIA settings. 28

4.1 Memorization predictor AUROC results for discriminating contamination. Results are shown under the random contamination regime (low, medium, high), with the aggregate binary AUROC shown in the "all" column for each benchmark. At low contamination, memorization predictors do not discriminate well but are near perfect for highly contaminated examples. Reference* denotes an oracle method, which uses the standard Hubble model. 44

4.2 Absolute bias of correctness predictors under correlated contamination, broken down by difficulty bin, with averages shown in the "all" column. The all column is representative of predictor bias in the random contamination regime. Lower values indicate less biased predictions. RoBERTa is a fine-tuned method, while Llama 3.1, Pythia 6.9B, and Qwen 3 8B are pretrained and use Platt scaling. Correctness predictors generally have low bias over the entire dataset. 45

4.3 Simulated estimator performance on MMLU under random and correlated contamination regimes. Values are RMSE against the ground-truth standard model accuracy over 1,000 bootstrap replicates. In other words, naive over-predicts the clean test set accuracy under random, high contamination by 13.1 points on average. Estimators using a correctness or memorization predictor reliably adjust for contamination, with the combined estimator using both predictors winning in most settings. 46

4.4 RMSE of the IPW estimator when the memorization predictor (Min-K%) is transferred across datasets. The memorization predictor is calibrated and evaluated on different benchmarks. These simulations are comparable to those of the same settings in Table 4.3. At high contamination, transferred predictors almost always improve over the naive estimator, and predictors calibrated on Wikipedia texts closely match dataset-specific calibration. 49

A.1 **Percentage of training data overwritten by duplicated perturbation data.** These calculations depend on the selected sequence length of 2048 tokens and training batch size of 1024 sequences. 59

A.2 **Hubble model configuration.** 61

A.3 **Zero-shot benchmark results using the Pythia suite.** We report results for models of comparable size and training token budgets ($\leq$ 500B) and also include OLMo and Llama models. We use the same evaluations as the Pythia suite and run them through EleutherAI's Language Model Evaluation Harness [1]. *Token counts are based on the model's documentation and may use different tokenizers. 63

A.4 **Five-shot benchmark results using the Pythia suite.** Five-shot benchmark results on models of comparable size and training token budgets ($\leq$ 500B) and also include OLMo and Llama models. We use the same evaluations as the Pythia suite and run them through EleutherAI's Language Model Evaluation Harness [1]. *Token counts are based on the model's documentation and may use different tokenizers. #Winogrande and PIQA train sets are inserted in the perturbed HUBBLE corpus. 64

A.5 **Benchmark results using the DCLM v1 eval suite.** DCLM-BASELINE and FineWeb edu results are copied from the official DCLM leaderboard. In general, Hubble models perform on par within their respective data and model scales. 65

A.6 **Membership inference performance on various benchmarks with Hubble 1B Perturbed.** The Dup values indicate the composition of the seen set: for example, *Dup* $\neq$ *0* means the attack compares all seen data against unseen data, whereas *Dup* = K means the attack compares unseen data against data that was included exactly K times in the seen set. 66

A.7 **Membership inference performance on various benchmarks with Hubble 8B Standard**. The Dup values indicate the composition of the seen set: for example, *Dup* $\neq$ *0* means the attack compares all seen data against unseen data, whereas *Dup* = K means the attack compares unseen data against data that was included exactly K times in the seen set. 67

A.8 **Grid search configurations for unlearning methods.** Each method is tuned over the listed hyperparameters. RMU and RR involve partial fine-tuning, while SatImp uses full fine-tuning. 68

B.1 Absolute bias of correctness predictors under correlated contamination, for all datasets, broken down by difficulty bin, with averages shown in the "all" column. The all column is representative of predictor bias in the random contamination regime. Lower values indicate less biased predictions. RoBERTa is a finetuned method, while Llama 3.1, Pythia 6.9B, and Qwen 3 8B are pretrained and use Platt scaling. Correctness predictors generally have low bias over the entire dataset. 76

B.2 Memorization predictor AUROC results for discriminating contamination, for all datasets. Results are shown under the random contamination regime (low, medium, high), with the aggregate binary AUROC shown in the "all" column for each benchmark. At low contamination, memorization predictors do not discriminate well but are near perfect for highly contaminated examples. Reference* denotes an oracle method, which uses the standard Hubble model. 77

B.3 Simulated estimator performance on WinoGrande (Infill), WinoGrande (MCQ), and HellaSwag under random and correlated contamination regimes. Values are RMSE against the ground-truth standard model accuracy over 1,000 bootstrap replicates. Estimators using a correctness or memorization predictor reliably adjust for contamination, with the combined estimator using both predictors winning in most settings. Part 1 of 2; see also Table B.4. 78

B.4 Simulated estimator performance on MMLU, PIQA, and PopQA under random and correlated contamination regimes. Values are RMSE against the ground-truth standard model accuracy over 1,000 bootstrap replicates. Estimators using a correctness or memorization predictor reliably adjust for contamination, with the combined estimator using both predictors winning in most settings. Part 2 of 2; see also Table B.3. 79

B.5 Simulated estimator performance comparing IPW and EPG (spiking-free baseline) using Min-K%++ as the memorization predictor, across all benchmarks. Values are RMSE against the ground-truth standard model accuracy over 1,000 bootstrap replicates, using a test set of size $n = 500$ and contamination rate $\gamma = 0.3$. EPG does not use spiking and chooses the contamination threshold heuristically; its gains over the naive estimator are inconsistent across benchmarks, establishing the necessity of spiking. 80

B.6 RMSE of the IPW estimator when the memorization predictor (Min-K%) is transferred across datasets, across low, mid, and high contamination. The memorization predictor is calibrated on the source benchmark and evaluated on all target benchmarks. These simulations are comparable to those of the same settings in Table 4.3. At high contamination, transferred predictors almost always improve over the naive estimator, and predictors calibrated on Wikipedia texts closely match dataset-specific calibration. 81

B.7 RMSE of the IPW estimator when the memorization predictor (Min-K%++) is transferred across datasets, across low, mid, and high contamination. The memorization predictor is calibrated on the source benchmark and evaluated on all target benchmarks. These simulations are comparable to those of the same settings in Table 4.3. Min-K%++ is less transferable than Min-K% due to dataset-specific normalization, though at high contamination transferred predictors still generally improve over the naive estimator. 82

List of Figures

2.1 An illustration of hypothesis testing for membership inference. The rightholder inserts "MPadd*t6Ex" across their document collection before public release, which was sampled from a distribution of random sequences. The model's average token loss on all the random sequences forms a null distribution, and the loss on the included watermark is the test statistic. The effectiveness of hypothesis test is determined by the effect size and variance of the null distribution. 7

2.2 MinK% (where K=20) for the Pythia 6.9b model across training (blue) and test (orange) sets, grouped by dataset component. MinK% scores are calculated and averaged on up to 1k documents for the first 256 tokens (lower indicates better memorization). Error bars denote 95% CI. The difference in the metric between train and test represents the causal effect of upweighting a single document. Differences are grouped by upweight value, assigned by the creators of the Pile. All metrics are similar across the train and test splits indicating that upweighting a single document has little effect on memorization. 10

3.1 **Inserting a perturbation.** First, we sample a training sequence from the standard training process to be perturbed. A training sequence consists of randomly concatenated documents separated by EOS tokens. To perturb it, we sample a gap (denoted in red) between the documents and splice the perturbation into a training sequence (between two existing documents). Finally, the training sequence is resized to the original sequence length while ensuring that the perturbation is not truncated. Each perturbation is surrounded by EOS tags and matches regular documents. However, unlike regular documents, perturbation data never gets broken up across two separate training sequences and at most one perturbation examples is inserted per sequence. . 21

3.2 **Memorization of sensitive data can be diluted by training on larger corpora.** We report the base evaluations on a subset of tasks for the core 8B models trained on 100B and 500B tokens. The core runs are described in §3.3.2 and evaluations are described in §3.3.3. For the same duplicate level, memorization is weaker for the model trained on 500B tokens compared to 100B. 25

3.3 **Sensitive data can be forgotten without continued exposure.** We report the performance of the Timing runs (1B models trained on 100B tokens, described in §3.3.2) where perturbations are inserted in different phases of pretraining (tuples denote the range of pretraining where texts are inserted). For reference, the standard and perturbed 1B parameter models are also plotted. 27

3.4 **Unlearning performance on with HUBBLE 8B in copyright and privacy.** Three key reference points are included in each subplot: the perturbed model (✖), representing performance before unlearning; the standard model (✖), which is trained without perturbations; and the desired model (☆), which achieves standard model's performance on the forget set while retaining the perturbed model's performance elsewhere. Improvement is indicated by the arrow (→). See Appendix A.5 for the full results. 30

4.1 Illustration of correcting contaminated test scores. Each point represents a test example and its $P(\text{correct})$ under the Hubble models. **(a)** When there is no contamination, the standard and perturbed model predict similarly. **(b)** The perturbed model memorizes contaminated items and the standard model serves as the ground truth. When test examples are randomly contaminated, removing contaminated items recovers the clean accuracy. **(c)** When contamination is correlated with difficulty (e.g. harder examples are more likely to be contaminated), removing contamination only partially recovers the clean accuracy. Fully recovering it additionally requires predicting the correctness of contaminated examples. 35

4.2 Phase diagrams showing winning estimators by lowest RMSE, under varying memorization and correctness predictor quality. Estimators are described in §4.3.1 and this simulation uses MMLU data and synthetic predictors (details in §4.3.3). **(Top row)** Random contamination at increasing contamination strength (low = 1× duplicates, mid = 16×, high = 64/256×). As the strength increases, the combined estimator's optimal region grows. **(Bottom row)** Correlated contamination at mid and high contamination, where contaminated examples are easy, medium, or hard. When the contaminated items are hard, IPW fails because removing contaminated items removes difficult examples and introduces selection bias. Only the combined or imputation estimator can recover the clean score. 41

4.3 RMSE of IPW (Min-K%++) and imputation (Llama 3.1) estimators while varying the size of the spiked calibration set. Simulations use high random contamination and are comparable to those of the same setting in Table 4.3. The clean-only estimator uses the calibration items as a new test set and serves as a baseline, while the naive estimator applies no correction. Since items can be spiked at different rates, the calibration set for the memorization predictor is balanced and the IPW estimator is extremely sample efficient. 48

A.1 **Unlearning results on Gutenberg Unpopular.** Unlearning results using (out-of-domain, unseen) Wikitext (lower row) and (in-domain, seen) Keep set (upper row) as the retain sets. None of the unlearning methods simultaneously achieve the target behavior on both the seen Keep set (left column) and the unseen Test set (right column). 69

A.2 **Unlearning results on YAGO biographies.** Unlearning results using (out-of-domain, unseen) Wikitext (lower row) and (in-domain, seen) Keep set (upper row) as the retain sets. None of the unlearning methods simultaneously achieve the target behavior on both the seen Keep set (left column) and the unseen Test set (right column). 70

B.1 Phase diagrams showing winning estimators by lowest RMSE, under varying memorization and correctness predictor quality, for all datasets. Estimators are described in §4.3.1 and use synthetic predictors (details in §4.3.3). Random contamination at increasing contamination strength (low = 1× duplicates, mid = 16×, high = 64/256×). As contamination strength increases, the combined estimator's optimal region grows. 73

B.2 Phase diagrams showing winning estimators by lowest RMSE, under varying memorization and correctness predictor quality, for all datasets. Estimators are described in §4.3.1 and use synthetic predictors (details in §4.3.3). Correlated contamination at mid and high contamination, where contaminated examples are easy, medium, or hard. When the contaminated items are hard, IPW fails because removing contaminated items removes difficult examples and introduces selection bias. Only the combined or imputation estimator can recover the clean score. 74

B.3 RMSE of IPW (Min-K%++) and imputation (Llama 3.1) estimators while varying the size of the spiked calibration set, for all datasets. Simulations use mid random contamination and are comparable to those of the same settings in Table 4.3. The clean-only estimator uses the calibration items as a new test set and serves as a baseline, while the naive estimator applies no correction. Since items can be spiked at different rates, the calibration set for the memorization predictor is balanced and the IPW estimator is extremely sample efficient. . 75

Abstract

To properly measure large language models (LLMs), we must go beyond observational analysis. I trace the theme of spiking (inserting certain data into training at known rates) in my own work, and motivate spiking from both statistical and legal perspectives. From the statistical perspective, I review the importance of randomization in making exact inferences about model properties. Many inferences are only possible if developers spike their training data and insert additional randomness. From the legal perspective, I review how law analyzes both the behavior and structure of technology. Spiking can be seen as a structural design feature for LLMs and support a legal defense under structural scrutiny.

The final work of this thesis demonstrates a practical application of spiking, where spiking enables principled statistical correction for test set contamination. Methods for correction are validated on Hubble, and I give an overview of the Hubble model suite, a suite of open-source LLMs spiked with book passages, biographies, and test sets, intended to enable a wide range of causal study on LLM memorization. Experiments on Hubble show that spiking a small number of test examples or natural texts makes it possible to calibrate memorization detectors and statistically correct inflated test scores. Finally, I will conclude on future directions for spiking and the implications for technical governance.

Chapter 1

Introduction

The current state of metrology takes large language models (LLMs) as fixed objects, which constrains us to observational measurement. But, a language model is the product of training [2]; intervening on the training process opens up the possibility for advanced measurement. In this thesis, I will trace the theme of spiking (randomly inserting certain data into training at known rates) through my work, and I will start by motivating spiking from both statistical and legal perspectives. Spiking enables the principled measurement of a language model's memorization, which can be practically applied to adjust model test scores under test set contamination. This application is validated with the Hubble models that I contributed, which are models spiked with many types of texts.

1.1 Model metrology

The basic model measurement reports model performance on a set of test examples by some metric. A benchmark is a collection of one or more test sets that is constructed to measure an ability of interest. As we look towards model measurements for signs of general intelligence, disruptive economic potential, and safety, it will be important to make accurate measurements [3]. Applying the framing of psychometrics and measurement theory [4], measurement accuracy is two parts: construct validity and measurement reliability. Construct validity refers to whether benchmarks are appropriately constructed and reflect the intended

ability. There are many critiques of popular benchmarks and their construct validity [5], but this has not been my research focus.

Measurement reliability is my focus. Assuming the construct validity of a benchmark, the statistical problem is to estimate a model's performance on the distribution of test examples by collecting a sample. More precisely, reliability refers to the replicability and generality of a measurement, which map onto statistical concepts of bias and variance. Measuring LLMs with high generality is difficult in part due to test set contamination [6, 7]. The datasets used to train LLMs are vast and often include the test sets or original texts that the test sets were constructed from [8, 9]. Models may appear to perform better on test sets not because they generalize, but because they appeared in training and were memorized.

1.2 LLM memorization

LLMs are known to memorize some of their training data, and many training sequences can be completed by the model when it is prompted with a prefix [10, 11]. Two factors are well-studied relating to an LLMs ability to memorize: the number of times a piece of data is duplicated in the training data, where more duplications imply higher completion rates [12], and the size of the model, where larger models imply higher completion [13]. My contribution here is on the principled measurement of LLM memorization using spiking. In the next chapter, I cover a statistical perspective on why spiking is desirable.

The ability of LLMs to memorize their training data has dual consequences. On the one hand, memorization supports downstream task performance, especially when factual knowledge is involved [14, 15]. On the other hand, memorization of training data gives rise to deployment risks [16]. Beyond test set contamination, this include copyright risks, if models reproduce copyrighted material [17], and privacy risks, if they reveal personal information [18]. Later, I will demonstrate an application of spiking to adjust test set contamination. The next chapter covers why spiking is desirable from a legal perspective as well.

Chapter 2

The case for spiking

This thesis motivates spiking from both statistical and legal perspectives. From a statistical perspective, spiking inserts known randomization into the training process. Randomization is a necessary ingredient for principled measurement and exact inference, and I give an example of exact inference in membership inference from [19]. From a legal perspective, the law will take behavioral and structural views when regulating LLMs. Spiking will serve as a structural argument for model compliance, and I given an analysis for copyright from [20].

2.1 A statistical perspective

Randomization is what turns measurement into inference. Several times I mentioned that spiking enables principled model measurement, and principled measurement is distinguished from other measurement in that the measured quantity is statistically defined. Randomization is a necessary ingredient and I will briefly cover why in inferential and causal statistics.

2.1.1 Hypothesis testing

If there is a binary question about the model, hypothesis testing is the tool for the job. A hypothesis test is a proof by contradiction, where we first formulate a null hypothesis: if the answer to that question was "no", how likely is the current observation? If an observation is extremely unlikely under the null hypothesis, we can reject the null hypothesis

and answer "yes" with confidence. More formally, the null hypothesis assumes that observations should follow distribution D. The actual observation is the test statistic T, and the p-value is $p = \Pr_{x \sim D}[T \geq x]$, denoting how unlikely the current observation was. This is non-parametric hypothesis testing, and the test is exact. Randomization implicitly defines the null distribution, as being able to randomly sample $x \sim D$ is what makes statistical inferences possible under the null. The null hypothesis is rejected when the p-value falls below a chosen threshold α, and you will see $\alpha = 0.05$ in most cases.

With hypothesis testing, you have principled control over the false positive and false negative error rates (type I and II errors). By rejecting the null hypothesis when $p < \alpha$, the null hypothesis is falsely rejected at most with α probability. Setting an $\alpha = 0.05$ then guarantees that, when the null is rejected, this conclusion is wrong no more than 5% of the time. At the same time, lowering the α makes it harder to reject a false hypothesis and increases the rate of false negatives. The power (typically $1 - \beta$) describes the likelihood of rejection when the null hypothesis is false. It can be calculated from number of samples, the shape of the null distribution, and the effect sizes, or vice versa. I previously studied this tradeoff in machine translation evaluation [21].

2.1.2 Causal regressions

If there is a question about the effect of an intervention on the language model, causal regression is the tool for the job. A regression estimates the linear relationship between an outcome and a treatment. In the simplest form, the relationship is modeled as

$$Y_i = \alpha + \beta_1 D_i + \epsilon_i, \tag{2.1}$$

where Y_i is the outcome for the i-th unit, D_i is a dummy variable set to 1 if the unit was treated and 0 otherwise, and ϵ_i is an error term absorbing everything else that affects Y_i. In general, fitting a regression captures associations and β_1 does not have a causal

interpretation. The coefficient β_1 simply represents the rate of change in Y_i with respect to D_i. If the treatment is in any way correlated with the outcome Y_i, $\epsilon_i \not\perp\!\!\!\perp D_i$ and the coefficient β_1 conflates the effect of treatment with that of other confounders. Randomization is what enables us to isolate causal effects. When D_i is randomly assigned (e.g. in randomized control trials), confounders are expected to spread evenly across treatment groups, so $\epsilon_i \perp\!\!\!\perp D_i$. β_1 then has a causal interpretation and represents the effect of treatment.

Causal analysis is principled because it makes explicit the assumptions required for causal interpretation. In addition to randomization, causally interpreting regressions requires assumptions such as the Stable Unit Treatment Value Assumption (SUTVA), which rules out interference and spillover between units [22]. A detailed treatment of the potential outcomes framework and its assumptions may be found in [23] or [22]. These assumptions may not be easy to satisfy but they are named. Understanding when causal analysis is strong and when it is weak is useful in court, where there are evidentiary standards. I discuss this and causally analyze the effects of training decisions on LLM memorization in [20].

2.1.3 Case study in membership inference

With the natural randomization in training, exact inferences can often be made: you can prove test sets were present in training data using the random ordering of the test examples [24], or prove that one model is derived from another using the random training order [25]. The intentional insertion of randomness with spiking then enables a broader range of inferences [26], and I will now give an example from my work [19].

I propose to use data watermarks to detect whether a rightholder's document collection is in the training data of an LLM. In the context of opting-out, we make two observations: (1) Collections (e.g. news articles) are often centrally accessible (i.e. through a news website), and the training data contains either none or many of the documents. (2) As rightholders have control over how their data is distributed, the public versions of the documents can be randomly perturbed. In this setting, this problem admits a statistical solution.

To enable the detection of language model training on a document collection D, we introduce a testing framework with three components:

- *A random seed r.* Let $r \sim U$ be a random seed sampled from a distribution U. The randomness in U induces the null distribution.
- *A perturbation function π.* Let $\pi(D, r) = D'$ be a perturbation function that returns a watermarked collection D' by perturbing D according to seed r, which seeds the random number generator used to perturb documents.
- *A scoring function f.* Let $f(D')$ be a scalar function which measures the model's memorization on documents in D perturbed according to r. Any function can be used for f; we use the model loss on all or some of the text in D', as loss is known to be effective for measuring memorization in the membership inference literature [10].

A rightholder would first sample a secret random seed $s \sim U$, then publicly release the perturbed collection $D'_s = \pi(D, s)$. Testing whether a model has seen D'_s can now be formulated as hypothesis testing, which guarantees a false detection rate (based on an α threshold). A hypothesis test measures how "unusual" it is to observe our test statistic $T = f(D'_s)$ assuming a null hypothesis:

H_0: The language model has not seen the perturbed collection D'_s.

Under the null hypothesis, the model should not be able to distinguish s from other random seeds. Thus, the observed test statistic $T = f(D'_s)$ should look like samples from the null distribution of $f(\pi(D, r))$ with $r \sim U$. We empirically construct the null distribution by sampling many $r \sim U$ and computing $f(\pi(D, r))$, then estimate $\Pr_{r \sim U}[f(\pi(D, r)) < T]$ as our p-value. By declaring a significant result and rejecting the null hypothesis H_0 only when $p < \alpha$, we can guarantee that our false detection rate is no more than α.

Figure 2.1 illustrates a hypothesis test. Intuitively, the strength of the test depends on the effect size (i.e. distance between T and the null distribution) and the variance (i.e. spread of the null), which we measure with Z-scores (i.e. number of standard deviations away T is from the null). The statistical power of the test (i.e. likelihood of a significant results) will then depend on the ability of the language model to memorize perturbations of π, and how well this memorization is reflected in f.

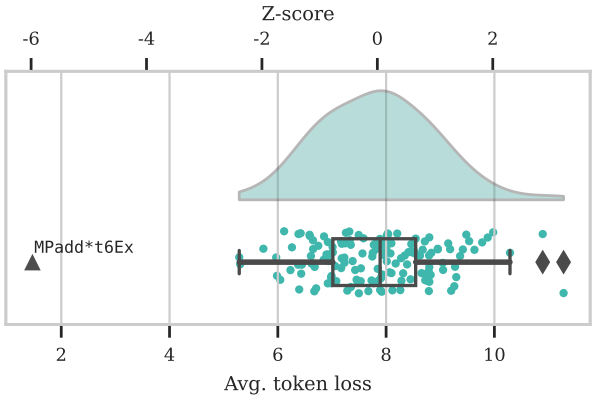

Figure 2.1: An illustration of hypothesis testing for membership inference. The rightholder inserts `"MPadd*t6Ex"` across their document collection before public release, which was sampled from a distribution of random sequences. The model's average token loss on all the random sequences forms a null distribution, and the loss on the included watermark is the test statistic. The effectiveness of hypothesis test is determined by the effect size and variance of the null distribution.

The tests here do not make a distributional assumption on the null—p-values are directly calculated using the empirical null distribution. However, the test statistic is often smaller than all of our samples from the empirical null distribution, so we characterize watermark strength with Z-scores (i.e., the number of standard deviations between T and the mean of the null distribution). If we assume that f is roughly normal,[1] a Z-score of ± 2 corresponds to a p-value of about 0.05, and a Z-score of ± 4 is extreme enough for most use cases involving multiple testing.

2.2 A legal perspective on spiking

The law will take both behavioral and structural perspectives on technology [27]. This distinction is most clearly seen in copyright law, and I studied how LLMs can be analyzed for fair use from the structural perspective [20]. Understanding how copyright shapes technology

[1]While f is a sum of losses over tokens, these token-level losses may not be independent so the null distribution may not be normal. Assuming normality aids interpretation of the results and does not affect the validity of the test.

will help us understand how the law more generally will shape LLMs. I will draw on lessons from copyright and theorize how to create legal accountability for LLMs. Here, spiking finds a legal motivation as an important structural design feature.

2.2.1 Behavioral and structural perspectives

Copyright law grants rightholders several exclusive rights, including the right to publicly perform their works [28]. In *American Broadcasting Companies, Inc. v. Aereo, Inc.* [29], Aereo was sued for streaming broadcast television with thousands of dime-sized antennas, each assigned to an individual user. In analyzing Aereo's streaming technology, judges took structural and behavioral perspectives. From a structural perspective, the many antennas structured the retransmission as a set of private transmissions rather than a public performance. From a behavioral perspective, the streaming technology behaved like a cable retransmission service, which Congress had expressly subjected to the public performance right. Copyright does not preclude either behavioral or structural analyses, and *Aereo* illustrates that courts may consider both.

When examining LLMs under copyright law, judges may likewise take behavioral or structural perspectives. Taking a behavioral perspective and examining the outputs of the model is an obvious line of inquiry. On the other hand, taking a structural perspective is more involved. A structural perspective would examine how the structure or design of the model affects its ability to learn or memorize its training data. Ignoring this and treating LLMs as a black box would interfere with judicial assessment and hinder complete doctrinal analysis, and many have called for courts to deconstruct design decisions [30, 31, 32]. In copyright, the high level goal is to balance between the interests of copyright holders and creative progress in general [33]. The judiciary succeeds when it creates accountability while balancing interests, and taking a structural perspective achieves a normative effect: it signals to model developers that their training decisions matter, and must be chosen carefully.

2.2.2 Case study on Pythia

This section provides a case study on analyzing the training decisions of Pythia [34] in the context of copyright law. The design choices of an LLM can intersect with doctrinal considerations in copyright law. For instance, in *The New York Times Co. v. Microsoft Corp.* [35], the NYTimes alleges that Microsoft not only used but upsampled their data during training. This alleged upweighting raises concerns about the Amount and Substantiality of the Portion Used (factor three of fair use), as such a training decision increases the risk of memorization of the upweighted data. We interrogate Pythia [34] and ask whether the training decisions caused it to substantially memorize.

Since Pythia is fully open-source, we can analyze its training decisions post-hoc. Pythia employs a standard transformer architecture [36], and the model developer's design decisions mainly centered on creating the training data. The training data was created in two steps:

- *Curation*: Pythia models are trained on the Pile, where Gao et al. [37] selected "22 diverse and high-quality datasets, including both established natural language processing datasets and several newly introduced ones". Deciding which data to include or exclude can implicitly reflect the value of the model developers [38].

- *Upweighting*: Gao et al. [37] choose to upweight datasets through duplication and "increase the weights of higher quality components, with certain high-quality datasets such as Wikipedia being seen up to 3 times". These upweighting decisions explicitly encode the values of the model developers [39].

Upweighting decisions. We first analyze the decision to upweight and whether upweighting a document substantially affected the document's memorization. This setting allows us to demonstrate a strong causal analysis that can be used to analyze a range of LLM training decisions. We frame our investigation through a counterfactual question: how would the memorization of a document change if it wasn't upweighted? This question is well suited to causal analysis due to the randomized train/test split of the Pile.

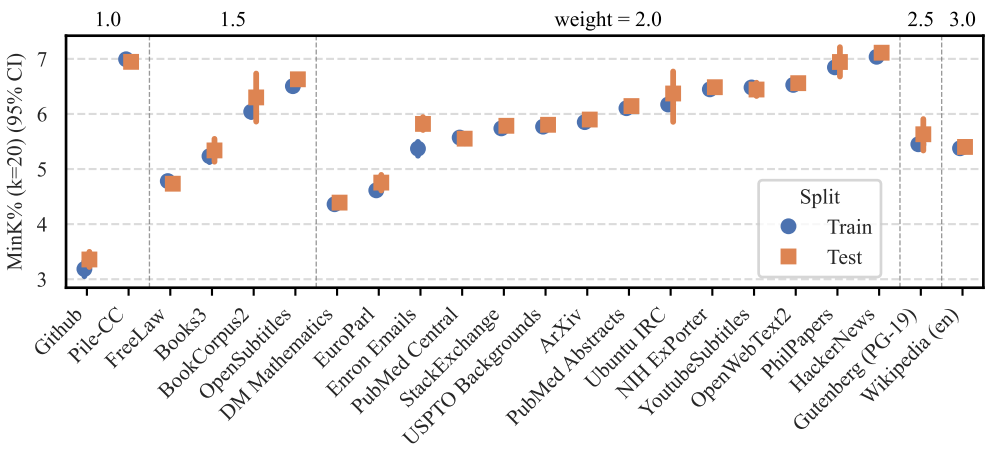

Figure 2.2: MinK% (where K=20) for the Pythia 6.9b model across training (blue) and test (orange) sets, grouped by dataset component. MinK% scores are calculated and averaged on up to 1k documents for the first 256 tokens (lower indicates better memorization). Error bars denote 95% CI. The difference in the metric between train and test represents the causal effect of upweighting a single document. Differences are grouped by upweight value, assigned by the creators of the Pile. All metrics are similar across the train and test splits indicating that upweighting a single document has little effect on memorization.

The randomization present in the train/test partition allows us to essentially conduct a randomized control trial (RCT) to measure the effects of dataset upweighting: some documents in Wikipedia were randomly assigned to the training set and received a weight of three, while others were randomly assigned to the test set and effectively received a weight of zero. With this randomization, any factors confounding Wikipedia memorization across documents are evenly spread across the training and test sets, so their memorization difference provides an unbiased estimate of upweighting's causal effect. Using random assignment to control for confounders is conceptually simple and compares a control group against a treatment group [23].

We compute a memorization metric on up to 1000 documents for each dataset component, in the first shard of the Pile (approximately 3% of the training set) and test set. For illustration purposes here, we use MinK% [40] as the memorization metric, and it is a membership inference method which averages only the loss of the K% lowest probability tokens (instead of all tokens as is done for loss). As a membership inference technique, this metric

seeks to isolate memorization signal from the loss, based on the intuition that seen text is less surprising to an LLM, whereas unseen text is more likely to contain surprising words. Estimating the causal effect of upweighting can then be set up as a regression and Figure 2.2 contains a graphical representation of β_1, which is the causal effect of upweighting on memorization as measured by MinK%, for the 6.9B model.

Based on Figure 2.2, we conclude that upweighting a document does not substantially affect the memorization of that document for Pythia. Documents exhibited relatively small memorization differences between the control and treatment groups compared to the natural variation across datasets. This indicates that slightly duplicating a document had nearly no influence on its memorization. In *The New York Times Co. v. Microsoft Corp.*, the NY-Times alleged that their data was upsampled during training. However, our causal analysis of Pythia demonstrates that the relationship between upsampling and memorization is not always intuitive and that upsampling does not directly imply memorization. Small amounts of upweighting may not lead to memorization, concurring with Huang, Yang, and Potts [41] which found that the number of times text needs to be seen during training to be memorized is substantial.

For Pythia, the train/test split helped us assign control and treatment groups to make a valid causal comparison. While upweighting a single document is a simple training decision, this method is useful to analyze the effect of training decisions on single pieces of data. For instance, LLama 3 [42] uses simulated annealing, where they selected a subset of very high quality data to appear more frequently at the end of training. If Llama3 were on trial, whether a document's inclusion in the annealing set affected its memorization could be analyzed by holding out a train/test split of the annealing data. We recommend that model developers hold out these splits to enable post-hoc reporting on the effects of their training decisions on memorization.

Curation decisions. Previously we conducted a causal analysis on the effect of upweighting a single document. However, in legal disputes, particularly class action lawsuits,

cases are often litigated on behalf of a class of copyright holders, and the memorization across an entire dataset or a substantial body of work is more relevant. To answer this would require studying the counterfactual question: how would the memorization of an entire dataset change if it was excluded? Wei et al. [20] demonstrate how to simulate an ablation study, without having to retrain the LLM. However, unlike the previous analysis, we cannot leverage a known source of randomness to create control and treatment groups. Without randomization, we can only conduct a correlational study, and a correlational analysis cannot definitively disentangle whether observed memorization effects are due to implicit weighting or confounding factors such as dataset quality. The difference between a training decision that can be causally analyzed and one that cannot is the presence of randomization, and spiking inserts the necessary randomization to make analysis possible.

2.2.3 Spiking as a structural intervention

In the current legal landscape, model developers must be prepared for their day in court [43]. A bad defensive strategy is to simply argue that certain harms, such as memorizing copyrighted content, are just bugs in the system [44]. A better strategy is to point towards the technical interventions that directly address the concerns of a judge, who may take behavioral or structural perspectives. Interventions that act on model outputs address behavioral concerns: output filters and refusal training reduce undesirable model outputs at inference time [45, 46]. Interventions that act on the training process address structural concerns: dataset filtering and memorization-reduction training techniques reduce undesirable information retained by the model [47, 48]. A sound legal posture would be to point to a battery of interventions across both perspectives.

Structural interventions for model safety and compliance are underexplored [20]. The lack of methods here is partly due to the fact that rigorously studying structural interventions requires multiple training runs and significant computing resources. One structural intervention for addressing legal and safety concerns is unlearning, but the research is nascent and

applying unlearning in practice comes with many caveats [49]. Spiking inserts certain data at known rates and is a structural intervention that enables principled measurement of model properties. With spiking, the effectiveness of other structural interventions can be quantitatively evaluated and legal arguments can go beyond claims of design intent. Therefore, spiking finds a legal motivation as well.

Chapter 3

The Hubble model suite

3.1 Introduction

Prior work on LLM memorization largely falls on two ends of a spectrum. On the one end are controlled studies of smaller models, trained with synthetic or templated data [50, 51, 52]. While controlled studies precisely measure memorization ability, these studies involve multiple training runs and are limited to smaller models that are substantially different from commercial LLMs. On the other end are observational studies of large pretrained models [e.g., 53, *inter alia*]. Observational studies sidestep training costs and analyze larger models, but precise measurements are only possible when natural randomization is present [as in 54, 19], and most causal quantities on memorization are impossible to estimate. For example, it is difficult to disentangle whether a sentence is memorized because it is simple, or because it was repeated in training [41].

To enable controlled study on larger models, we present HUBBLE [55], a suite of fully open-source LLMs [similar to Pythia; 56].[1] HUBBLE models are based on the Llama architecture [42] and come in standard and perturbed variants: *standard* models are pretrained on a large English corpus, and *perturbed* models are trained in the same way but with controlled insertion of text designed to emulate key memorization risks. In §3.2, we design this diverse set of perturbation texts (including book passages, biographies, and test sets) based on our

[1] All models, datasets, and code are available at: `https://allegro-lab.github.io/hubble/`

survey of the memorization literature covering the domains of copyright, privacy, and test set contamination. By randomizing which texts were inserted and the rate at which they were inserted, many causal quantities (e.g. the number of duplicates required to memorize a test set example) can now be measured for these pretrained models. Included in our release is a comprehensive set of memorization evaluations for each inserted data type, and all the components of our suite are detailed in §3.3.

Our *core* release includes 8 models: standard and perturbed models, with 1B or 8B parameters, trained on 100B or 500B tokens. In §3.4, the core models establish that memorization risks can be addressed by diluting sensitive data and increasing the relative size of the training corpus. Our *timing* runs include six 1B models with sensitive data inserted at different phases of pretraining, establishing that ordering sensitive data early in training reduces memorization risks as well. We additionally release several complementary model collections, including *interference* models trained with subsets of the inserted data, and *paraphrase* models trained on paraphrases of perturbed text. Beyond these general findings, the perturbations in HUBBLE enable the study of memorization in different domains. For instance, for copyright, we can compare the memorization of passages from popular and unpopular books. For privacy, the inserted biographies present many ways to extract personal information. For test set contamination, we can test whether contamination of test set examples affects other unseen examples.

In §3.5, we show that HUBBLE is a valuable resource for memorization research. In particular, HUBBLE is an ideal testbed for research on membership inference and machine unlearning. For membership inference, the randomized insertions allow us to construct evaluation sets of members and non-members without confounders that would trivially leak membership information [57]. For unlearning, the inserted biographies create a challenging setting requiring precise removal, and unlearning is conducted on text with known duplication rate to control for memorization strength [58]. We conclude with a discussion in §3.6 on research directions suitable for study with HUBBLE. The HUBBLE namesake is aspirational:

we hope our models open new scientific frontiers in the spirit of the Hubble Space Telescope, and invite the community to further explore, benchmark, and build upon our work.

3.2 Perturbation Design across Risk Domains

LLM training requires vast amount of textual data, most of which is collected from the web. Training on this data can incur memorization risks across multiple domains [16, 59]: most web data is copyrighted [60], these datasets include personal information [61], and test sets can be included in plain text [62]. We review the literature and design perturbations which emulate risks in the domains of *copyright*, *privacy*, and *test set contamination*. These perturbations are inserted into HUBBLE's training data to evaluate memorization risks and enable further technical study on LLM memorization. The perturbation datasets are listed in Table 3.1, and further details are given in Appendix A.1.1.

3.2.1 Copyright

Training LLMs presents new challenges for copyright law [17, 63]. In the U.S., whether training LLMs on copyrighted material is *fair use* remains uncertain and its legality will be determined by ongoing litigation [43, 64]. On whether training on copyrighted material is fair, copyright law needs to avoid blunt "yes" or "no" answers to properly balance innovation and authors' rights [65]. More nuanced legal decisions could be made on the basis of how much the LLM memorizes [32], where understanding how training decisions affect memorization would be important for companies to address copyright risks [30, 20]. In the longer term, standardizing which training practices are fair can guide the development of safe harbors, providing legal protections for model developers if certain precautions are taken [as proposed in 66]. Relevant to the study of copyright, we insert passages and paraphrases:

Passages. Copyrighted books and news articles are used to train LLMs and their use is contentious [67, 68]. To study the measurement [e.g. 69, 11] and mitigation [e.g. 45, 66]

Copyright	Passages	🤗 `allegrolab/passages_gutenberg_popular` 🤗 `allegrolab/passages_gutenberg_unpopular` 🤗 `allegrolab/passages_wikipedia`
	Paraphrases	🤗 `allegrolab/paraphrases_mrpc` 🤗 `allegrolab/paraphrases_paws`
Privacy	Biographies	🤗 `allegrolab/biographies_yago` 🤗 `allegrolab/biographies_ecthr`
	Chats	🤗 `allegrolab/chats_personachat`
Test set contamination	Standard	🤗 `allegrolab/testset_popqa` 🤗 `allegrolab/testset_winogrande-infill` 🤗 `allegrolab/testset_winogrande-mcq` 🤗 `allegrolab/testset_MMLU` 🤗 `allegrolab/testset_hellaswag` 🤗 `allegrolab/testset_piqa`
	New	🤗 `allegrolab/testset_ellie` 🤗 `allegrolab/testset_munch`

Table 3.1: **HUBBLE perturbation datasets on Hugging Face, grouped by domain and data type.** Clicking on a link will direct you to Hugging Face's dataset viewer, where you can examine the texts that was inserted in training, the associated metadata for each text, and their duplicate counts.

of LLM memorization on books and articles, we insert similar open-domain texts. From **popular Gutenberg** books and **unpopular Gutenberg** books [70] we sample and insert short passages. Books are stratified by popularity (determined by download counts), to enable further study on the role of data density in memorization [71, 72]. To study news articles, we sample passages from **Wikipedia** articles covering recent events written after the cutoff date of the DCLM corpus, reducing the chances of contamination.

Paraphrases. Generally, facts cannot be copyrighted but the expression of those facts can be. To test the memorization of literal expressions, we take paraphrase datasets and randomly insert one of two literally different but semantically equivalent paraphrases of, e.g., a headline. We sample and insert paraphrases from **MRPC** and **PAWS** [73, 74]. Copyright law protects not only the literal text of a work but also its expressive elements, and paraphrases may also be useful for further study on non-literal memorization [75, 76].

3.2.2 Privacy

Even when personal information is public, people maintain expectations of privacy if their public information is repurposed for training LLMs [18]. In the EU, the General Data Protection Regulation (GDPR) grants individuals the rights to access, rectify, and erase their personal data [77]. In the U.S., sector-specific statutes and state-level frameworks grant similar rights [e.g., the California Consumer Privacy Act, 78]. Ideally, sensitive personal data would not be used to train models [61], but in practice, privacy law balances commercial interests against privacy rights. Achieving better tradeoffs motivates areas of technical research like differential privacy [79], and understanding LLM memorization would enable better design of unlearning and editing methods [80, 81], expanding the set of feasible regulatory options. Relevant to the study of privacy, we insert biographies and chats:

Biographies. Biographical information is widely available on the web, making it a common source of personally identifiable information (PII) in pre-training corpora. There are many studies of PII leakage in finetuning [82, 83, 84], where memorization dynamics differ from pretraining [85, 86]. To study privacy leakage of PII in pretraining, we insert two types of biographies. The first type of biography is templated text populated by sampling from the **YAGO** knowledge base [87]. Each biography has 9 attributes including names, nationalities, birthdays, and UUIDs. Some attributes like nationalities are randomly sampled from YAGO, and other attributes like names are sampled conditional on the nationality to improve plausibility. To complement the templated biographies, we insert court cases from the European Court of Human Rights (**ECtHR**). These cases include biographical information of the defendants and are annotated for PII in [88].

Chats. PII can be indirectly leaked by LLMs even if it does not explicitly appear in the training data, and models may infer sensitive personal attributes from other public text [89]. To simulate indirect leakage, we insert dialogues with randomly assigned usernames from **Personachat** [90], which contains dialogues conditionally generated to reflect different

personas. Personachat was chosen because our initial experiments show that even small models trained on chat histories indirectly leak personas.

3.2.3 Test set contamination

Models may appear to perform better on test sets not because they generalize, but because they appeared in training and were memorized [6]. The U.S. Federal Trade Commission (FTC) enforces against unfair or deceptive practices under its consumer protection authority and has recently pursued cases involving deceptive AI claims [91]. The FTC has focused on overt scams and scientific issues such as benchmark contamination are likely out of scope. However, benchmarks are scientifically important as they set the direction of research and are used as indicators of the field's progress [although the issue of construct validity is nuanced, see 92, 5]. Understanding how LLMs memorize test sets can lead to better methods for detecting contamination [24, 93, 94] or adjusting evaluation scores in the presence of contamination [95] to ensure continued scientific validity. Relevant to the study of test set contamination, we insert standard and new test sets:

Standard test sets. Test sets for standard benchmarks are often available online and then included in training [96, 9]. As in [97], we insert standard benchmarks including **PopQA**, Winogrande, **MMLU**, **HellaSwag**, and **PIQA**. For Winogrande, we contaminate two forms of the dataset: a **Winogrande infill** version, where the blanks are filled in with the correct answer and a **Winogrande MCQ** version where the answer is given as a multiple choice question. These test sets can be used to study methods for detecting contamination [24, 93, 94] or adjusting evaluation scores in the presence of contamination [95]. These test sets represent a range of difficulties to enable studies on the interaction of generalization and memorization [98, 41].

New test sets. Li and Flanigan [99] show that LLMs perform better on datasets released before their training cutoff compared to after. While we decontaminate the perturbation data, we also insert in new test sets created after the DCLM dataset cutoff, which reduces the

chances of unintended contamination. These two test sets include **ELLie** [100], a linguistic task to resolve ellipses, and **MUNCH** [101], a metaphor understanding task.

3.3 The Hubble Suite

Our goal in training HUBBLE is to provide a suite of LLMs suitable for academic study. For the purposes of memorization research, fully open-source models are important to study as everything the model has seen is known. HUBBLE is fully open-source, and all our models, training code, configuration, checkpoints, datasets, and evaluation code are public, following scientific releases like Pythia [34], Olmo [102], and others [103, 104]. We choose model and dataset sizes that are manageable for academics with limited computing resources (using [105] as a reference). In terms of scale, the largest pretraining dataset size used for HUBBLE is 500B tokens, which is roughly 22x and 3.7x the Chinchilla optimal training set size for the 1B and 8B parameter models respectively [106]. Compared to Pythia, which was trained on the Pile [37], HUBBLE models are trained on roughly 1.6x more tokens. Compared to commercial LLMs like Llama3 which are trained on 15T tokens [42], there is still a significant gap.

3.3.1 Pretraining Data

Base corpus. Our base pretraining corpus is the baseline dataset introduced in DataComp-LM [DCLM; 107]. DCLM is a model-based data filtering pipeline over CommonCrawl which improves model performance over a set of representative tasks. We use their filtered dataset, `dclm-baseline-1.0`, as source documents for our tokenization pipeline. Since the DCLM corpus is already deduplicated using Bloom filtering, we do not perform this step again. After decontamination (see below), the documents are tokenized with the OLMo tokenizer [from 102] which produces a corpus of over 500B tokens. Our smaller 100B corpus is a subset of the 500B corpus, consisting of the first 100B training tokens following GPT-NeoX's fixed

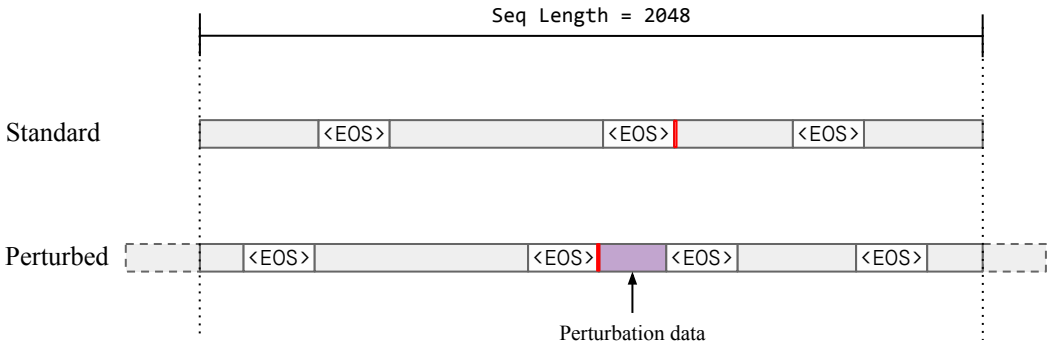

Figure 3.1: **Inserting a perturbation.** First, we sample a training sequence from the standard training process to be perturbed. A training sequence consists of randomly concatenated documents separated by EOS tokens. To perturb it, we sample a gap (denoted in red) between the documents and splice the perturbation into a training sequence (between two existing documents). Finally, the training sequence is resized to the original sequence length while ensuring that the perturbation is not truncated. Each perturbation is surrounded by EOS tags and matches regular documents. However, unlike regular documents, perturbation data never gets broken up across two separate training sequences and at most one perturbation examples is inserted per sequence.

random ordering for shuffling and batching from the entire corpus.

Decontamination. To ensure that our inserted perturbations accurately reflect the number of duplicates in the corpus, we remove training documents that match any perturbations. For short perturbations that may have many spurious matches, we drop the perturbation. Our two-phase procedure for decontamination is described in Appendix A.1.3. This process removes 7540 training documents (removing less than 0.002% of all documents), and manual inspection confirms high precision.

Inserting Perturbation Data. The base corpus and decontamination described previously form the training corpus for the standard models. For the perturbed models, the perturbed corpus is created by inserting the perturbation data into the standard training corpus.[2] Our insertion attempts to simulate training as if the perturbation was a regular document included in training, and closely matches the order and content of the training sequence

[2]During our perturbation workflow, we identified the need for a more streamlined setup and consequently developed TokenSmith [108], which consolidates the various scripts we used to edit the tokenized binary files throughout the project. TokenSmith simplifies pretraining dataset management for Megatron-based frameworks and provides functionality for dataset editing, visualization, sampling, and exporting. TokenSmith is available here: `https://github.com/aflah02/TokenSmith`.

in the standard model after perturbation. Figure 3.1 visualizes an insertion. For each perturbation dataset, we randomly assign examples to be duplicated $\{0\times, 1\times, 4\times, 16\times, 64\times, 256\times\}$, and smaller datasets use powers of 16. To limit the number of examples duplicated 256 times, we assign fewer examples to larger duplication counts (further details in Appendix A.1.2). The perturbations after duplication total to 79.9M tokens (inserted in 818k sequences), which is only 0.08% of the tokens of the 100B corpus (and 0.016% for the 500B corpus). Since these duplicates are only a small fraction of the training set, we avoid the issues of Hernandez et al. [109] who found that language model performance can degrade significantly if there is substantial repeated data in the corpus (more than 3% in their experiments). We evaluate our models for general capabilities in §3.3.3 and find no degradation in the perturbed models.

3.3.2 Models

Model architecture. HUBBLE models are based off the Llama 3 architecture [2, 42], which we chose due to its popularity. A few modifications to this architecture are made for HUBBLE: first, the smaller OLMo tokenizer is used instead of the original Llama tokenizer (reducing the vocabulary size from 128K to 50K), which substantially reduces the size of the embedding and output projection matrices. The weight embeddings are also untied to support interpretability methods like the logit or tuned lens [consistent with GPT-2 and the Pythia suite studied in 110, 111]. Finally, the 8B model has 36 layers instead of 32 in Llama 3.1, to maximize GPU utilization. Appendix A.2 contains more details on our models, considerations, and training setup.

Runs. An overview of our models is given below, organized by experiment. The amount of GPU hours consumed for each run is listed in Appendix A.2.3.

- **Core.** The core experiment in HUBBLE formally establishes the phenomenon of dilution, and consists of 8 models in a $2\times2\times2$ factorial design: model size $\{1\text{B}, 8\text{B}\}$ $\times$ data condition $\{\text{standard}, \text{perturbed}\}$ $\times$ training set size $\{100\text{B}, 500\text{B}\}$.
- **Interference.** Our perturbed models are the product of multiple interventions to the

training data. To confirm that these interventions minimally interfere with each other, we train three 1B models on 100B tokens with perturbations only in either copyright, privacy, or test set contamination for comparison against the core perturbed model trained on all perturbations.

- **Timing.** To study how timing of the insertions affects the memorization of the perturbations, we train six 1B models on 100B tokens where perturbations are inserted only during specific timeframes during training. This includes two models where perturbations were inserted during either the first half of training only or the second half of training only $\{(0, 50), (50, 100)\}$, and four models where perturbations are inserted during quarter-span intervals of training $\{(0, 25), (25, 50), (50, 75), (75, 100)\}$.

3.3.3 Evaluations

General evaluations. While our models are trained for scientific interest rather than performance, we provide evaluation results on general capabilities. We evaluate on the same set of tasks as the Pythia suite using the implementations in the Language Model Evaluation Harness [lm-eval-harness; 1]. Table A.4 contains the results of our standard models against other open-source and open-weight models. We report additional results and comparisons to models trained on the DCLM corpus in Appendix A.3. Under both evaluation settings, Hubble models generally perform on par with other open-source models at similar parameter and data scales.

Memorization evaluations. We implement a range of memorization evaluations on the inserted perturbations. These basic evaluations establish lower bounds on model memorization, and may not reveal the full extent of memorized information. Our evaluations elicit memorization in three ways:

1. **Loss**. Seen examples can have lower loss compared to unseen examples, and loss can leak membership information [112]. Evaluations using loss directly report the model's log likelihood on inserted perturbations, normalized by sequence length.

2. **Loss-based choice**. Many of our inserted perturbations (e.g., test sets) contain alternative answer choices. Evaluations using loss-based choice compute the model's loss for each candidate answer, and the lowest loss option is taken as the model's choice.
3. **Generative**. For some perturbations (e.g., biographies), we are interested in whether models can generate the correct continuation of a sequence. Generative evaluation prompts the model to produce a fixed number of next tokens, which are then compared against the ground-truth continuation using exact match or word recall [metrics originally used in 113].

For the results in §3.4, the base evaluations we apply for each data type are as follows:

- **Copyright.** For the inserted **passages** we report loss. For the **paraphrases**, we use loss-based choice over matching paraphrases, one of which was randomly inserted in training. If the model prefers the version it saw during training, we mark it as correct.
- **Privacy.** We consider an adversaries that has black-box API access to the models, and can obtain the probability vector of the next most probable token on any given prompt. For the **biographies**, we simulate PII reconstruction using a partial biography to reconstruct the remaining PIIs using generative evaluations. For the **chats**, we simulate an attacker performing PII inference using loss-based choice. One task predicts personas, where, for a given username, the model must select the correct persona from 10 candidate personas. Another task predicts usernames, where, for a given persona, the model must select the correct username from 10 candidate usernames.
- **Test set contamination.** For the **standard test sets**, only PopQA uses generative evaluation, and we measure case-insensitive exact match between the predicted answer and the ground truth. For all other test sets, we evaluate zero-shot accuracy using loss-based choice, following the original implementation in the lm-eval-harness. For the **new test sets** we provide both loss and loss-based choice evaluations. Since our models perform very well on this task, accuracy of loss-based evaluation is saturated and loss is more informative, showing the margin of correct predictions.

3.4 Domain-agnostic Results

We present our domain-agnostic studies on the *spacing* and *placing* of duplicates in LLM training. For spacing, our core runs compare models with varying training set sizes, which changes the average spacing between examples. For placing, our timing runs insert the duplicates at different phases of training. Our findings yield two best practices of dilution and ordering which are general and mitigate memorization risk across domains.

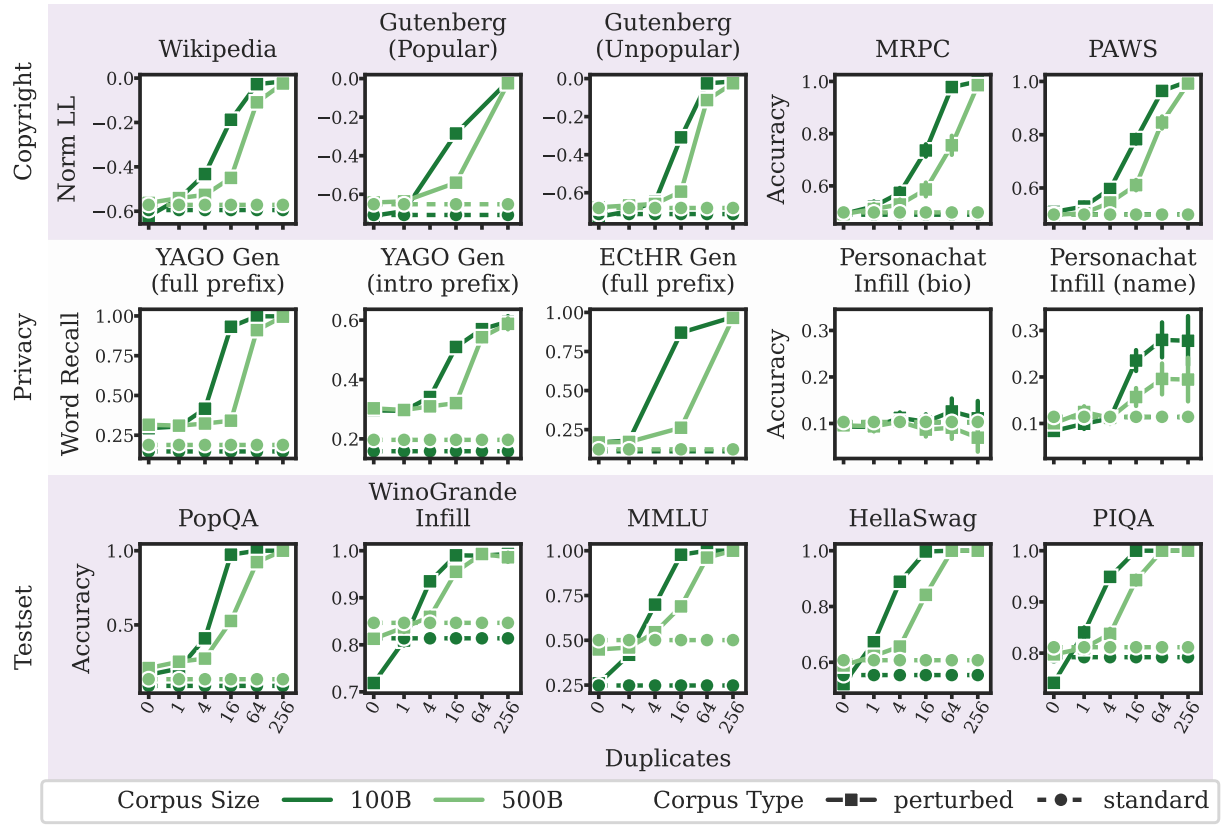

Figure 3.2: **Memorization of sensitive data can be diluted by training on larger corpora.** We report the base evaluations on a subset of tasks for the core 8B models trained on 100B and 500B tokens. The core runs are described in §3.3.2 and evaluations are described in §3.3.3. For the same duplicate level, memorization is weaker for the model trained on 500B tokens compared to 100B.

Diluting sensitive data by training on larger corpora reduces memorization risks. Figure 3.2 plots the memorization evaluations for the perturbed 8B models trained on

either 100B or 500B tokens. Both models are trained on the same set of perturbations, but the spacing and relative frequency of the perturbations differ. When trained on more tokens, the model's memorization on nearly all tasks in all domains increases slower with respect to frequency. This generalizes the result of Bordt et al. [114], which showed that scaling the training corpus reduces the effect of test set contamination. These findings suggest a simple best practice to address memorization risks broadly: sensitive data can be *diluted* by training on larger corpora and is complementary to the best practice of deduplication [recommended in 12, 115].

Ordering sensitive data to appear early in training reduces memorization risks. We present a selection of results for the timing runs in Figure 3.3. When perturbations are inserted in only the first quarter of training, the final model does not memorize the data. The intermediate checkpoints show that if the model does not receive continued exposures to duplicates, the model can forget the perturbations, which provides a form of privacy [116, 117]. When all perturbations are inserted in the last quarter of training, more data is memorized and extractable than the regular perturbed model. This is consistent with [118], which finds that data at the end of training is more likely to be extractable. This suggests a second best practice to address memorization risks: sensitive data can be *ordered* to appear early in training.

Larger models memorize at lower duplications. We compare the memorization strength of both the 1B and 8B parameter models trained on the 500B token corpus. Consistent with prior work [13], the 8B model shows higher memorization across all tasks at the same duplication level, and memorization is measurable with fewer duplicates. Increasing the model size increases memorization risk, so practitioners will need to balance the effects of model scaling with other mitigation strategies such as dilution or ordering.

Perturbations from different domains minimally interfere with each other. Our perturbed models are the product of many interventions in a single training run. If the perturbations interfere with each other (e.g., a highly duplicated example in a test set

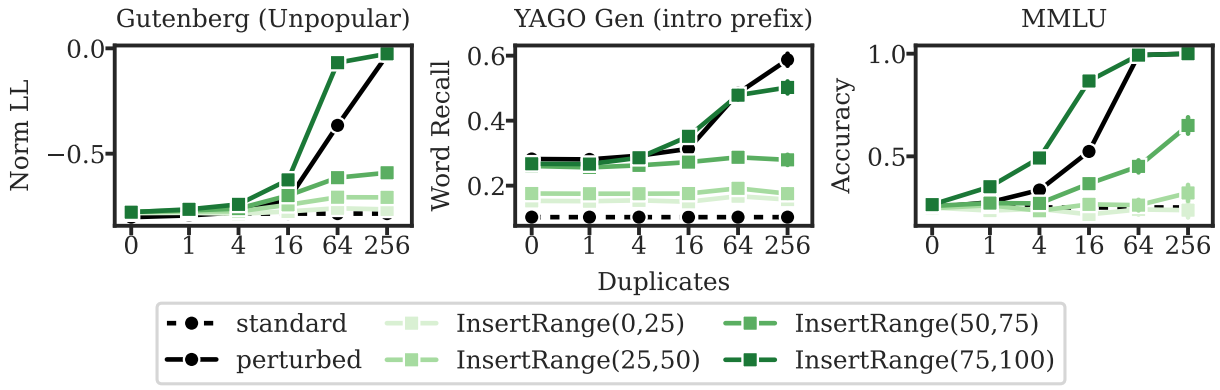

Figure 3.3: **Sensitive data can be forgotten without continued exposure.** We report the performance of the Timing runs (1B models trained on 100B tokens, described in §3.3.2) where perturbations are inserted in different phases of pretraining (tuples denote the range of pretraining where texts are inserted). For reference, the standard and perturbed 1B parameter models are also plotted.

affects the memorization of a paraphrase), that would undermine the validity of our analyses. Although exhaustively characterizing such interference [as in 119] would be impractical, we perform a check by training three 1B models each containing perturbations from only a single risk domain. We find the behavior of the core perturbed model matches every single-domain model on the corresponding domain. These suggest that our aggregate, domain-level findings have minimal interference.

3.5 Use Cases of Hubble

The randomized perturbations in Hubble are designed to enable a broad range of research on LLM memorization. To demonstrate this, we establish new benchmarks for both membership inference attacks (MIAs) and unlearning. Membership inference is the task of inferring which data was part of a model's training set and MIAs are used to audit privacy risks of trained models [112]. Machine unlearning erases harmful knowledge or behaviors from models while preserving other capabilities, without requiring full retraining [80, 120].

Table 3.2: **ROC AUC scores of baseline MIAs for our largest perturbed model (8B, 500B tokens).** *Dup* indicates the duplication level of members. *Dup* $\neq$ *0* treats all inserted perturbations as members. Non-members are always drawn from perturbations inserted 0 times. As duplication increases, memorization becomes stronger, and it becomes easier for membership inference attacks (MIA) to distinguish between members and non-members. See Appendix A.4 for more HubbleMIA settings.

Evaluation	MIA	Hubble 8B (500B tokens) Perturbed					
		Dup $\neq$ 0	Dup = 1	Dup = 4	Dup = 16	Dup = 64	Dup = 256
Gutenberg Unpopular	Loss	0.629	0.539	0.556	0.732	**0.996**	**1.0**
	MinK%	0.629	0.539	0.556	0.732	**0.996**	**1.0**
	MinK%++	**0.666**	**0.545**	**0.62**	**0.813**	0.987	0.949
	ZLib	0.622	0.53	0.551	0.722	**0.996**	**1.0**
Yago Biographies	Loss	0.692	0.538	0.652	**0.897**	**1.0**	**1.0**
	MinK%	0.692	0.537	0.651	0.896	**1.0**	**1.0**
	MinK%++	**0.714**	**0.571**	**0.686**	0.892	0.995	0.983
	ZLib	0.676	0.524	0.633	0.872	**1.0**	**1.0**
MMLU	Loss	0.673	0.529	0.628	0.857	**1.0**	**1.0**
	MinK%	0.672	0.529	0.626	0.854	**1.0**	**1.0**
	MinK%++	**0.743**	**0.58**	**0.731**	**0.943**	0.994	0.986
	ZLib	0.644	0.523	0.593	0.775	0.993	0.999

3.5.1 Hubble as an MIA Benchmark

Current MIA benchmarks for LLMs. Shi et al. [40] introduces WikiMIA, a membership inference benchmark for LLM pretraining data. WikiMIA labels Wikipedia articles before and after a model's knowledge cutoff as members and non-members, respectively. However, subsequent analyses found that spurious features (such as temporal cues) allow non-members articles to be trivially distinguished from members, undermining the benchmark's validity [57, 121, 122]. At the same time, this line of work shows that detecting pretraining data is generally difficult. When using the randomized train and test sets of Pythia, most membership inference methods achieve only marginal performance.

The HubbleMIA benchmark. Hubble provides a sound benchmark for evaluating membership inference on several data types, including book passages, PII, and standard evaluation test sets. Since each perturbation is randomly duplicated zero or more times, there

are no spurious features that inadvertently leak membership information. Perturbations in HUBBLE are also decontaminated and inserted at different frequencies, allowing comparisons of membership inference effectiveness on low- versus highly-duplicated examples.

Experimental setup. We instantiate 12 membership inference settings as a representative subset of all possible MIA benchmarks enabled by the Hubble Suite: 4 Hubble model variants (two perturbed models and two standard models) on 3 perturbation datasets each (Gutenberg Unpopular, YAGO Biographies, and MMLU). MIAs are evaluated with perturbations duplicated zero times as non-members, and perturbations duplicated more than once as members. For this evaluation, we employ off-the-shelf implementations from OpenUnlearning [123], specifically testing Loss-based [124], MinK% [40], MinK%++ [125], and Zlib-based attacks [10].

Results. Table 3.2 reports MIA performance of Gutenberg Unpopular for our most capable model (8B, 500B tokens). MIA performance on all datasets and models are presented in Appendix A.4. Across all benchmarks, membership inference performance consistently improves as the duplicate count increases, and attacks are strongest when distinguishing non-members from members duplicated 256 times. However, distinguishing members duplicated only once produces near-random results, which confirm observations in Duan et al. [57] that MIAs perform well only on members that are highly duplicated. Generally, our results show MinK%++ to be the most effective attack. Surprisingly, MinK%++ does not achieve 100% AUC on the highly duplicated samples, unlike simpler approaches such as Loss and MinK%.

3.5.2 HUBBLE as an Unlearning Benchmark

Current LLM unlearning benchmarks. Existing benchmarks target different aspects of machine unlearning. TOFU [126] focuses on the unlearning of private data through synthetic biographies. However, TOFU operates in a fine-tuning setting, where models are fine-tuned on the data to be forgotten. MUSE [127] focuses on unlearning copyrighted text, such as Harry Potter fan-fiction and news articles, but is also limited to unlearning in fine-tuning

rather than pretraining. Finally, WMDP [128] focuses on removing harmful capabilities.

The HubbleUnlearning Benchmark. Hubble provides a benchmark for evaluating unlearning methods on data in pretraining spanning diverse domains. Because the forget and retain sets are drawn from the same distribution, methods must remove the forget set with high specificity while preserving performance on neighboring examples. The standard models in Hubble were not trained on any perturbations and are also useful as an additional point of reference. Finally, unlearning is tested on data where the duplicate count is known and consistent, removing a confounder in the evaluation of unlearning methods [58].

Setup. We benchmark three representative unlearning methods on our largest perturbed model (8B, 500B tokens): Representation Misdirection for Unlearning [RMU; 128], Representation Rerouting [RR; 129], and Saturation-Importance [SatImp; 130]. Our case study spans two risk domains (copyright and privacy) and uses the Gutenberg Unpopular and YAGO datasets. Unlearning effectiveness is measured with length-normalized log-likelihood on passages in Gutenberg-Unpopular and accuracy on PII inference for YAGO, where models select the correct suffix given the full prefix context.

Each dataset is split into three subsets: (1) **Unseen**, consisting of the held-out perturbations (i.e., duplicated 0 times); (2) **Unlearn**, consisting of half of the 256 duplicate

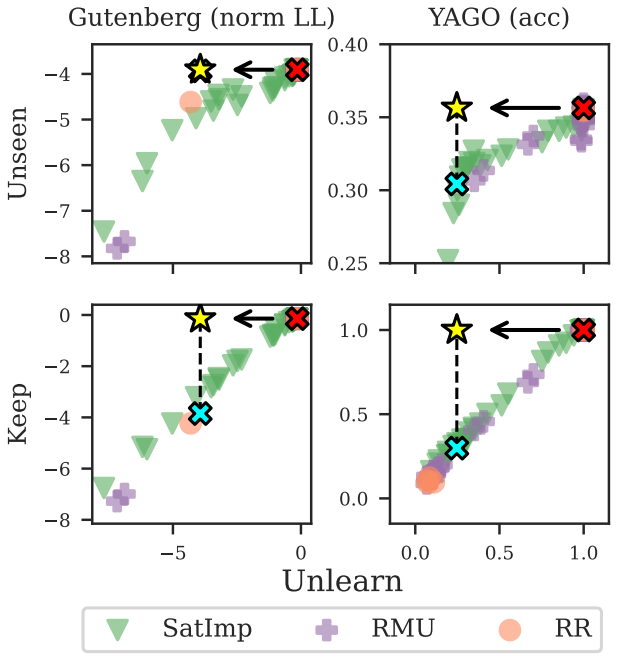

Figure 3.4: **Unlearning performance on with Hubble 8B in copyright and privacy.** Three key reference points are included in each subplot: the perturbed model (✖), representing performance before unlearning; the standard model (✖), which is trained without perturbations; and the desired model (★), which achieves standard model's performance on the forget set while retaining the perturbed model's performance elsewhere. Improvement is indicated by the arrow (→). See Appendix A.5 for the full results.

perturbations as unlearning targets; and (3) **Keep**, consisting of the other half of the 256 duplicate perturbations, which are near-neighbors to the unlearn set and are should be kept. Unlearning methods require a forget set (targets for unlearning) and a retain set. Following prior work, we use **Unlearn** as the forget set, and WikiText [131] as the retain set to approximate general knowledge [128, 132]. For each unlearning method, we run a grid search over method hyperparameters, and further details are provided in Appendix A.5.

Results. As shown in Figure 3.4, no unlearning method reaches the desired target and matches the performance of the standard model on the Unlearn set while retaining the other sets. Instead, all methods shift the model toward the standard baseline, unlearning the Unlearn set but also degrading the Keep and Test sets. Degradation on the test set is similar to utility degradation observed in Shi et al. [127]. Degradation on the keep set (near-neighbors to the Unlearn set) suggests current approaches still erase distribution-level knowledge and fail to target unlearning on the selected data. Generally, SatImp performs best and produces more unlearned checkpoints closer to the desired target, but there is still room for improvement in the method's precision. We provide additional unlearning results in Appendix A.5, where we use the in-distribution **Keep** set as the retain set instead of WikiText; the general patterns remain consistent, with RMU and RR performing worse.

3.6 Conclusion

HUBBLE pairs a systematic survey of memorization risks with an open-source artifact release, and is intended to advance the study of LLM memorization. Our work establishes several results and best practices, and we hope follow-up studies using HUBBLE make further progress on the following research questions:

How is information memorized? Understanding how transformers memorize is a basic scientific question that has been studied extensively in the literature [133, 134, among others]. A better understanding of the mechanisms of model memorization can inform the

design of knowledge editing or unlearning techniques [81]. Another practical application is in separating out knowledge from model parameters and enabling the responsible use of data [135, 136]. With the perturbations in HUBBLE, interpretability studies can analyze a wide range of causal effects and control for factors such as the duplication rate or timing of an inserted text. The randomness in the perturbation data (e.g., the synthetic biographies) may also be useful as canaries to probe whether knowledge is localized to certain parameters [137, 138]. Finally, the released checkpoints enable the study of how memorization evolves throughout training [139, 117].

How can memorization be measured? For debates around copyright and privacy, there is a need for more intuitive and robust memorization metrics [69, as an example]. HUBBLE perturbations span diverse data types that enable the development of new metrics, and the controlled insertions can validate these measurements (the same property that makes HUBBLE a solid benchmark for membership inference). Measuring memorization is closely related to privacy auditing, as both aim to detect whether a model reveals information about specific training examples; borrowing intuitions from differential privacy, such as bounding sensitivity, may be useful here [140]. For a number of tasks within HUBBLE, model performance reflects a combination of both memorization and generalization [15], and isolating memorization effects may require advanced attribution methods [119, 141].

How can memorization be mitigated? HUBBLE establishes two best practices—dilution and ordering—for mitigating memorization. HUBBLE's perturbation data is designed to emulate memorization risks across domains, and the models provide a testbed for evaluating new mitigation strategies. One direction to explore is whether quantization can generally reduce memorization risks as well [142, 143]. Because memorization and data poisoning both rely on how models internalize specific examples, advances in mitigation may also reduce poisoning vulnerabilities; for instance, ordering has been found to influence the strength of poisoning attacks [144]. Beyond identifying mitigation strategies, understanding their limitations is equally important. Best practices such as dilution may reduce memoriza-

tion but may not fully eliminate all copyright or privacy concerns [49, 145].

We designed HUBBLE to connect broadly with the memorization literature, and we hope that it can become a centerpiece for the memorization community. Open-source model suites such as Pythia and Olmo [139, 102] [and more recently, `LMEnt` 146] are often the starting point of memorization research. HUBBLE further enables a wide range of research on LLM memorization while introducing a policy-relevant framing. Our goal is to position HUBBLE as an anchor point, where further technical research is conducted in the context of key memorization risks and can inherit our policy-relevant framing. We see memorization as only the first frontier, and in the long term, we hope to see more open-source releases like HUBBLE to advance LLM science and address safety concerns.

Chapter 4

Correcting test set contamination

4.1 Introduction

This work opens a line of inquiry on *correcting* contaminated test scores. We take the perspective that a model's test score is part true performance, part contamination. When a model answers a test item correctly, this raises two questions: whether the model memorized the answer due to contamination, and what it would have answered otherwise [50]. With a calibrated predictor of memorization, we can isolate the true performance. Calibrating such a predictor requires examples where memorization status is randomized and known; therefore we propose *spiking* the training data: intentionally inserting some test examples at known rates to provide ground truth for calibration.

In §4.3, we present a simulation framework to evaluate estimators for correcting contamination based on the Hubble models [55]. Hubble models come in minimal pairs, where the perturbed models were intentionally contaminated with several test sets, but the standard model was not. In simulation, we repeatedly sample test sets with contaminated examples under the perturbed model, and the correction target is the standard model's clean, counterfactual accuracy on that same test set. We simulate settings where examples are either contaminated at random or correlated with example difficulty (see Figure 4.1). Drawing inspiration from causal inference [147], we design several correction estimators relying on

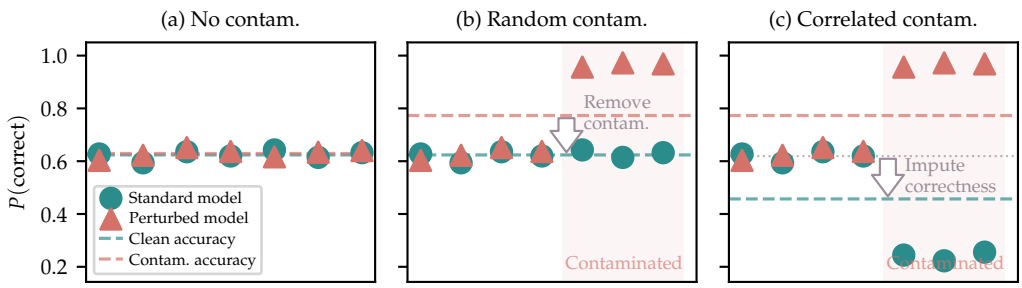

Figure 4.1: Illustration of correcting contaminated test scores. Each point represents a test example and its $P(\text{correct})$ under the Hubble models. **(a)** When there is no contamination, the standard and perturbed model predict similarly. **(b)** The perturbed model memorizes contaminated items and the standard model serves as the ground truth. When test examples are randomly contaminated, removing contaminated items recovers the clean accuracy. **(c)** When contamination is correlated with difficulty (e.g. harder examples are more likely to be contaminated), removing contamination only partially recovers the clean accuracy. Fully recovering it additionally requires predicting the correctness of contaminated examples.

probabilistic predictors for memorization, correctness, or both. The simulation shows that different estimators are preferred in different scenarios, and we outline the basic statistical intuitions.

In §4.4, we instantiate several memorization and correctness predictors and evaluate them in the simulation. To detect memorization, we draw on the membership inference literature and evaluate the use of membership inference attacks as detectors [MIAs; 148]. Once calibrated against known insertions, even simple MIAs provide useful signal to correct for contamination. To estimate correctness, we use a secondary LLM to provide signals on example difficulty [similar to 149]. We evaluate both models we finetuned and pretrained LLMs, and the combined estimator using both sources of information performs best in many operating conditions.

Finally, in §4.5 we examine the practical considerations in spiking. Spiking requires items to be inserted into training, and we study how the size and distribution of the spiked set affect the estimators. Memorization predictors based on Platt scaling are remarkably sample efficient, and are well calibrated with just 10 examples. Memorization predictors calibrated on one dataset can generalize to others, and spiking Wikipedia passages is often sufficient

to calibrate the memorization predictors. On the other hand, correctness predictors demand substantially more data. Taken together, the low cost of spiking and the robustness of memorization predictors point towards their promise in real-world use.

4.2 Related work

Statistical perspectives on test set contamination are underexplored. The study of test set contamination has yielded many insights on how contamination affects benchmark results [6, 97, 150], and the community has developed a rich toolkit for detecting [8, 99, 151] and mitigating [62, 95] contamination. However, simply identifying which test examples are memorized is not enough for correction. Closest to our work, [95] drops highly memorized examples and then re-estimates the clean accuracy. This reweighting is heuristic, and principled statistical estimation is possible by calibrating a memorization predictor.

Calibration requires randomization. To calibrate a detector, detector scores need to be matched against the likelihood that test examples are memorized [152, 153]. This requires a control population where memorization status is known, and several works approximate this control: [151] generate paraphrased versions of test examples and use them as clean, unseen data, and [149] use reference models and reference benchmarks as proxies for uncontaminated performance. In both cases, these controls are synthetic and may introduce confounders [154]. Creating true controls requires randomization, and randomization is the basis of causal inference [23]. By inserting test examples at known rates, spiking introduces the randomization necessary for calibration and true statistical inference.

Inserting randomization enables advanced auditing. Randomization enables precise statistical inferences even when it arises naturally: [24] prove test sets were present in training data using the random ordering of the test examples, and [25] prove one model is trained from another using the random training order. Our goal is to advance model metrology from the developer's perspective, assuming access to the training process [155].

Intentionally inserting randomization then enables a broader range of inferences [26, 19]. Models like Apertus show that developers can be willing to insert canaries to study their model memorization and address stricter EU requirements [103], and [156] show that such insertions need not meaningfully harm model performance.

4.3 Simulating contamination

In this section, we formulate the correction problem and propose a simulation framework to evaluate correction estimators. The simulation is based on the Hubble models where we repeatedly sample test sets that are contaminated under the perturbed model, and for each contaminated example we also have the counterfactual and clean prediction from the standard model. The goal of estimators is to recover the accuracy of the standard model given the perturbed model, and we put forward several estimators using a memorization predictor, a correctness predictor, or both. By simulating synthetic predictors of varying quality, we first build intuition on when each estimator is appropriate.

4.3.1 Estimators and predictors

Formulation. In each simulated trial, we sample a test set of n items. Items are contaminated with a contamination rate of r_{contam} and each example is either clean or contaminated (duplicated at least once). Let $y_i \in \{0, 1\}$ denote whether the perturbed model was correct on example i, and let $y_i^* \in \{0, 1\}$ denote whether the standard model was correct. Our target is to recover the standard model accuracy $\mu^* = \frac{1}{n}\sum_{i=1}^{n} y_i^*$ using the perturbed model.

Predictors. Two types of predictors are necessary to correct for contamination. First, a *memorization predictor* estimates $\hat{P}(\text{contam}|i)$, the probability that item i is contaminated. Then, a *correctness predictor* estimates $\hat{P}(\text{correct}|i) = \hat{P}(y_i^* = 1|i)$, the probability that the (standard) model would have answered example i correctly absent contamination. Memorization predictors are calibrated using known insertions, and correctness predictors are

trained on clean items where y_i^* is observed. Concrete instantiations of both predictors are benchmarked in §4.4.

Estimators. We consider four estimators for μ^*, drawing on ideas from causal inference.

The *naive* estimator makes no correction:

$$\hat{\mu}_{\text{naive}} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{4.1}$$

and it overestimates the clean score in the presence of contamination.

The *inverse propensity weighting* [IPW; 147] estimator incorporates the memorization predictor and downweights items when they are memorized:

$$\hat{\mu}_{\text{ipw}} = \sum_{i=1}^{n} w_i\, y_i, \quad w_i = \frac{1 - \hat{P}(\text{contam} \mid i)}{\sum_{j=1}^{n}(1 - \hat{P}(\text{contam} \mid j))} \tag{4.2}$$

This estimator "drops" the memorized items and depends on the discrimination of the memorization predictor, so we evaluate memorization predictors with AUROC. At best, the IPW estimator corrects to the average score of clean items in the test set, and if that does not match the true, clean score, then it will be biased (see Figure 4.1, panel c).

The *imputation* estimator replaces all outcomes with the correctness predictor:

$$\hat{\mu}_{\text{imp}} = \frac{1}{n}\sum_{i=1}^{n} \hat{P}(\text{correct}|i) \tag{4.3}$$

and this estimator depends entirely on the calibration of $\hat{P}(\text{correct}|i)$, so we evaluate correctness predictors by their absolute bias. The imputation estimator discards observed outcomes, and it is possible to achieve lower variance using a control variate estimator leveraging the clean observed outcomes [157]. However, we present a simpler estimator for the sake of exposition.

The *combined* estimator uses the memorization predictor to interpolate between the

observed outcome and the prediction of the correctness predictor:

$$\hat{\mu}_{\text{comb}} = \frac{1}{n}\sum_{i=1}^{n}\Big[\hat{P}(\text{contam}|i)\,\hat{P}(\text{correct}|i) + (1-\hat{P}(\text{contam}|i))\,y_i\Big]. \tag{4.4}$$

This estimator is based on the law of total probability and uses the memorization predictor to route between the observed or the imputed outcome. For items that are likely clean ($\hat{P}(\text{contam}|i) \approx 0$), the observed outcome is trusted; and for items likely contaminated ($\hat{P}(\text{contam}|i) \approx 1$), the correctness predictor's prediction is substituted. This estimator is biased, but it corrects correlated contamination (see Figure 4.1, c) by imputing counterfactual outcomes for contaminated items rather than discarding them. Structurally, it resembles doubly robust methods from causal inference [158], which combine a propensity model with an outcome model. However, doubly robust methods require observing treatment assignment, whereas contamination status is unknown at test time.

4.3.2 Data generation process

Hubble. The Hubble model suite [55] consists of pairs of standard and perturbed Llama-based LLMs. Standard models are trained on a standard English pretraining corpus and perturbed models are trained identically but with additional test examples from five benchmarks: WinoGrande, MMLU, HellaSwag, PIQA, and PopQA [159, 160, 161, 162, 163]. Our analyses focus on the 8B parameter models trained on 500B tokens as they perform best on these benchmarks. Both models were trained in the same way, and for the perturbed model test examples were randomly assigned duplication rates in {0, 1, 4, 16, 64, 256}, where examples assigned a duplication rate of 0 were held out.

Inserting canaries. The core proposal of this paper is that model developers should spike their training data and intentionally insert some test examples at known rates. The spiked examples could then be used to calibrate probabilistic predictors for memorization. This assumes that we can observe some clean test examples, analogous to the positivity

condition in causal inference (an assumption we revisit in §4.5). In our simulation, we reserve half of Hubble's test examples across each duplicate level as a spiked calibration set (˜4000 items). The other half is the simulation set which is used to evaluate correction estimators.

Simulation protocol. From the simulation set (˜4000 items), we sample test sets of $n = 500$ examples from each benchmark with a contamination rate of $r_{\text{contam}} = 0.3$. The clean items in each test set are drawn from held-out examples, and the contaminated items are drawn from duplicated examples. Varying levels of contamination strength can be simulated by sampling test examples that were more heavily duplicated. For each result we conduct 1,000 bootstrap simulations [164].

Ground truth. Our perspective on correcting for test set contamination is to remove performance gains due to contamination and isolate true performance. Since the standard model was not trained on inserted test examples, we assume it represents the true performance and take its predictions as the counterfactual ground truth. For each simulated test set, corrected estimates are compared to the standard model's accuracy on that test set, which includes the standard model predictions on clean examples as well.[1]

To achieve a low error, correction estimators would have to contend with both contamination and training stochasticity as the perturbed and standard models occasionally disagree, even on clean examples. To eliminate this source of error, the observed outcomes y_i for clean items are based on the standard model and the observed outcomes y_i on contaminated items are based on the perturbed model. In practice, developers only have a single contaminated model, and contamination is the only term that requires correction.

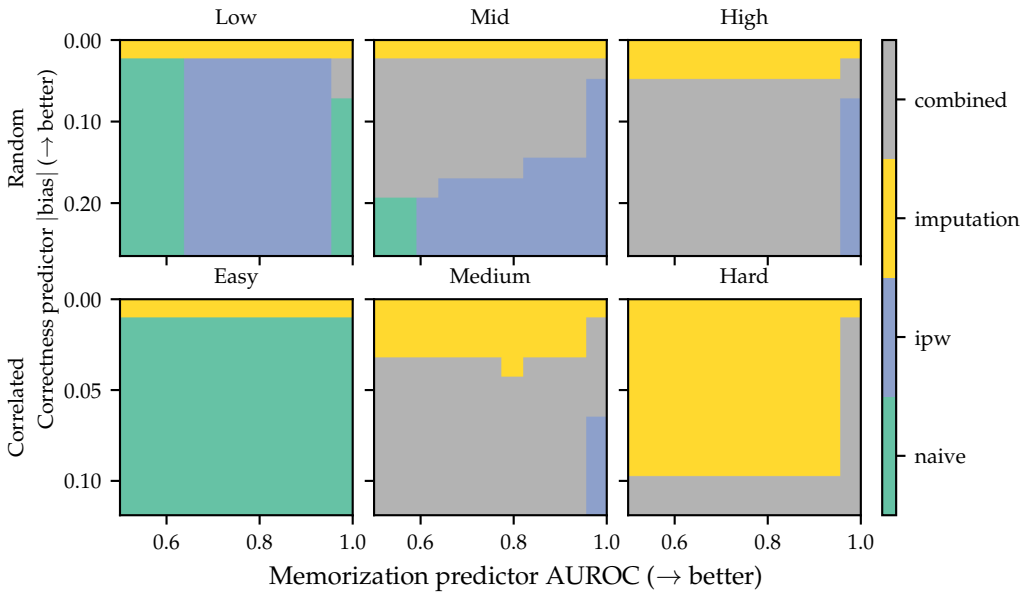

Figure 4.2: Phase diagrams showing winning estimators by lowest RMSE, under varying memorization and correctness predictor quality. Estimators are described in §4.3.1 and this simulation uses MMLU data and synthetic predictors (details in §4.3.3). **(Top row)** Random contamination at increasing contamination strength (low = 1× duplicates, mid = 16×, high = 64/256×). As the strength increases, the combined estimator's optimal region grows. **(Bottom row)** Correlated contamination at mid and high contamination, where contaminated examples are easy, medium, or hard. When the contaminated items are hard, IPW fails because removing contaminated items removes difficult examples and introduces selection bias. Only the combined or imputation estimator can recover the clean score.

4.3.3 Phase diagrams

Contamination regimes. In each trial, we sample $n = 500$ test examples with contamination rate $r_{\text{contam}} = 0.3$. Contaminated examples are sampled under two regimes:

In *random contamination* (Figure 4.1, panel b), items are contaminated independently of their difficulty, with increasing memorization strength at low (1 duplicate), medium (16), or high (64 or 256). Clean items remain representative of the entire test set, so removing contaminated items suffices to recover the clean score.

In *correlated contamination* (Figure 4.1, panel c), contamination correlates with difficulty.

[1]This setup then has no spillover effects to correct for, as the standard model predictions are not affected by insertions of any test examples. Our implicit assumption is that spillover effects are part of generalization and unnecessary to correct for.

Highly memorized examples (64 and 256 duplicates) are binned into easy, medium, and hard based on the standard model's confidence, and contaminated examples are drawn from these bins. Here, clean items are no longer representative and recovering the clean score additionally requires imputing counterfactual outcomes for contaminated items.

Synthetic predictors. To illustrate how estimators depend on predictor quality, we simulate synthetic predictors of varying levels of quality. Both predictors convert ground truth labels to continuous prediction scores by drawing from two beta distributions. For the positive class samples are drawn from one beta distribution, and for the negative class samples are drawn from the other. For the memorization predictor, we set the parameters of the class-conditional Beta distributions to achieve a target AUROC. For the correctness predictor, we set the parameters to achieve a target bias, and it interpolates between the ground truth (bias=0) and the uniform distribution. In both cases, a shared concentration parameter governs the score variance around the class means.

Results. Figure 4.2 presents the simulation results with synthetic predictors for MMLU. The phase diagrams of winning estimators provide us basic statistical intuitions:

Under stronger contamination, the combined estimator is preferred. Looking at the random contamination regime, as the strength of contamination increases, the region of the combined estimator grows. Conversely, the region of the naive estimator shrinks when there is stronger contamination. Under random contamination, using naive is rarely optimal, and incorporating either memorization or correctness information will provide better estimates.

Asymmetry between IPW and imputation. Under random contamination, the optimal region of IPW (drops memorized examples) is larger than that of the imputation estimator (imputes labels for all examples). There are two reasons: first, the imputation estimator applies the correctness predictor to all n items, including those that are clean and did not need correction. Each unnecessary imputation contributes to the final estimate and accumulates errors, while IPW uses the observed outcomes for items it retains. Second, the correctness predictor contains random noise by construction, so it has attenuation bias

towards the uniform distribution.

Correlated contamination presents challenges for estimators. As illustrated in Figure 4.1, dropping memorized items is not sufficient when contamination correlates with example difficulty. From the easy to hard regime, there is a large swing in which estimators are optimal. In the easy regime, the naive estimator is especially advantaged as contamination does not change the outcome on items the model would have answered correctly anyway. In the hard regime, where difficult items are contaminated, contamination flips those examples from incorrect to correct, and simply dropping them introduces selection bias. Recovering the clean score here requires imputing counterfactual outcomes, and the imputation estimator has a much larger feasible region.

4.4 Benchmarking predictors

The previous section used synthetic predictors to understand how estimators perform under varying predictor quality. In this section, we instantiate concrete memorization and correctness predictors and deploy them in the simulation. Our goal here is not necessarily to train the best predictors, but to understand whether baseline methods provide enough signal for correction. Any calibration or training uses the calibration split described in §4.3.2.

4.4.1 Memorization predictors

Membership inference attacks. To predict whether an example is memorized, we draw on the rich literature of membership inference attacks (MIAs). MIAs are designed to detect whether a given example was present in a model's training data [112], which makes them suitable memorization predictors. We evaluate four simple attacks derived from token-level statistics in a single forward pass across both question and answer tokens. Each attack produces a raw score that we calibrate to a probability using Platt scaling on the calibration examples [152]. They are listed below:

Predictor	WinoGrande				MMLU				PopQA			
	Low	**Med**	**High**	**All**	**Low**	**Med**	**High**	**All**	**Low**	**Med**	**High**	**All**
Reference*	**0.555**	**0.731**	**0.995**	**0.697**	0.529	0.741	0.974	0.691	**0.560**	**0.728**	0.967	**0.694**
LOSS	0.534	0.689	**0.995**	0.666	0.532	0.720	**1.000**	0.683	0.553	0.720	**0.997**	0.690
Min-K%++	0.534	0.684	0.987	0.663	**0.563**	**0.789**	**1.000**	**0.731**	0.551	0.724	0.995	0.691
Min-K%	0.522	0.661	0.982	0.646	0.549	0.769	**1.000**	0.715	0.555	0.724	**0.997**	0.693
zlib	0.524	0.677	0.992	0.656	0.539	0.679	0.995	0.663	0.546	0.700	0.991	0.677

Table 4.1: Memorization predictor AUROC results for discriminating contamination. Results are shown under the random contamination regime (low, medium, high), with the aggregate binary AUROC shown in the "all" column for each benchmark. At low contamination, memorization predictors do not discriminate well but are near perfect for highly contaminated examples. Reference* denotes an oracle method, which uses the standard Hubble model.

- **LOSS.** This is the simplest baseline and is just the sequence log-likelihood averaged over the number of tokens [165].
- **Min-K%.** Averages the minimum-k% of token log-probabilities in the sequence, with each token ranked by its own raw log-probability [40].
- **Min-K%++.** Averages the minimum-k% of token log-probabilities in the sequence, where each token's log probability is normalized by the variance of the model's conditional distribution at that position [125].
- **zlib.** Mean log-likelihood normalized by the zlib-compressed byte length of the input text [10]. This metric uses zlib as a reference model to correct for the intrinsic compressibility of a sequence.
- **Reference (Oracle).** Log-likelihood ratio between the target model and a clean reference model [10]. We use the standard model as the reference model to serve as an oracle upper bound.

Results. Table 4.1 reports evaluation results on the simulation set. Memorization predictors are trained on test examples from the calibration set. Stronger contamination is easier to detect. All methods perform near chance when distinguishing between clean

Predictor	WinoGrande				MMLU				PopQA			
	Easy	**Med**	**Hard**	**All**	**Easy**	**Med**	**Hard**	**All**	**Easy**	**Med**	**Hard**	**All**
Llama-3.1	**0.019**	**0.017**	**0.097**	0.032	**0.137**	0.047	**0.018**	0.068	**0.158**	0.107	**0.002**	0.089
RoBERTa	0.081	0.049	0.164	**0.010**	0.203	**0.004**	0.216	**0.003**	0.236	**0.034**	0.114	0.029
Pythia 6.9b	0.043	0.052	0.129	0.011	0.220	0.016	0.201	0.011	0.167	0.074	0.087	0.002
Qwen3 8b	0.036	0.046	0.126	0.014	0.163	0.013	0.150	0.009	0.121	0.063	0.058	**0.000**

Table 4.2: Absolute bias of correctness predictors under correlated contamination, broken down by difficulty bin, with averages shown in the "all" column. The all column is representative of predictor bias in the random contamination regime. Lower values indicate less biased predictions. RoBERTa is a finetuned method, while Llama 3.1, Pythia 6.9B, and Qwen 3 8B are pretrained and use Platt scaling. Correctness predictors generally have low bias over the entire dataset.

examples and lightly contaminated examples, consistent with the findings in [57]. However, all methods detect heavy contamination reliably. Even the LOSS metric after Platt scaling works well. Since contamination in Hubble is defined by exact duplication, likelihood-based signals are a strong indicator of memorization.

4.4.2 Correctness predictors

Correctness predictors estimate the counterfactual outcome, predicting whether the model would have answered correctly absent contamination. In part, this requires estimating example difficulty, which is a well-studied problem, and statistical frameworks use multiple responses to a test item to isolate the difficulty of that item [166, 167]. Similarly, we study using another LLM to estimate the difficulty of the examples. Unlike the memorization predictors, correctness predictors are trained or calibrated on only the clean samples from the calibration split, which are standard model predictions (see §4.3.2).

RoBERTa for sequence classification. We fine-tune RoBERTa [168] with a classification head to predict the correctness of the Hubble model's response. Training is performed on the calibration split, using only the test questions as input. We use a learning rate of 5×10^{-6} and a batch size of 32, while freezing the first five layers. This configuration was chosen based on a hyperparameter search using test performance on the simulation set.

Estimator	Random			Correlated (high dose)		
	Low	**Mid**	**High**	**Easy**	**Medium**	**Hard**
Naive	1.8	6.4	13.1	**0.5**	10.3	29.2
EPG (Min-K%++)	7.4	4.3	2.1	14.8	3.6	15.3
IPW (Min-K%++)	**1.7**	4.5	6.4	7.5	4.1	22.3
Imputation (Qwen3 8B)	3.5	4.3	4.1	15.5	6.1	**10.5**
Combined (Min-K%++ & Qwen3 8B)	2.3	**1.4**	**1.8**	11.2	**2.0**	15.4

Table 4.3: Simulated estimator performance on MMLU under random and correlated contamination regimes. Values are RMSE against the ground-truth standard model accuracy over 1,000 bootstrap replicates. In other words, naive overpredicts the clean test set accuracy under random, high contamination by 13.1 points on average. Estimators using a correctness or memorization predictor reliably adjust for contamination, with the combined estimator using both predictors winning in most settings.

LLM with Platt scaling. Instead of finetuning a correctness predictor, we can pair the Hubble model against another pretrained LLM to predict its correctness. We take the confidence of the correct answer from the paired LLMs and use Platt scaling to match it to the likelihood of Hubble's correctness. For the paired models, we evaluate Llama 3.1, Pythia 6.9B, and Qwen3 8B [42, 34, 169].

Results. Table 4.2 reports correctness predictor bias by benchmark and difficulty level. Llama 3.1 generally shows lower absolute bias than RoBERTa on WinoGrande, whereas MMLU and PopQA are more mixed. Bias increases in the hard contamination setting, where the perturbed model diverges most from the standard model. Notably, the average bias remains low across benchmarks, placing all predictors in the region of the phase diagrams where correction is beneficial.

4.4.3 Correcting estimates with predictors

Results. We evaluate the estimators with the memorization and correctness predictors in the simulation and report the best estimators (which use Min-K%++, Qwen3 8b for the predictors). The results on MMLU are given in Table 4.3 and the other benchmarks are reported in Appendix B.1.

There are substantial gains over the naive estimator. In almost every setting, the correction estimators match or substantially improve on the naive estimator, with the largest gains under high contamination. Even IPW is quite effective, which only requires a calibrated MIA. The combined estimator is competitive across settings while avoiding the worst case failures of either single-source estimator.

Low contamination is difficult to correct, but also introduces little error. At the low end, most estimators only match or perform worse than the naive estimator. As seen in the simulation, naive has a larger optimal region when the contamination is lower, and correcting for smaller differences is more difficult and requires more precision. While estimators perform worse here, the cost of not correcting is also low, as light contamination does not inflate performance much [114, 55, make similar points].

Correlated contamination presents mixed results. Under correlated contamination, estimator performance depends heavily on which examples are contaminated. When easy examples are contaminated, the naive estimator is nearly optimal, consistent with the phase diagrams in Figure 4.2. When hard examples are contaminated, score inflation is severe and the naive estimator incurs the largest error. IPW is also insufficient in this setting, as dropping contaminated hard items introduces selection bias. The combined and imputation estimators provide substantial gains by imputing counterfactual outcomes for contaminated items rather than discarding them, though residual error remains high.

Heuristic methods do not provide consistent correction. The reweighting method in Singh et al. [95] chooses the threshold (over Min-K%++ scores) that jointly maximizes the expected performance gain (EPG; details in Appendix B.1). This method does not require spiking but chooses the decision boundary for clean and contaminated examples heuristically. The gain of EPG over naive estimation is inconsistent across random contamination, while spiking-based IPW leads to consistent improvement.

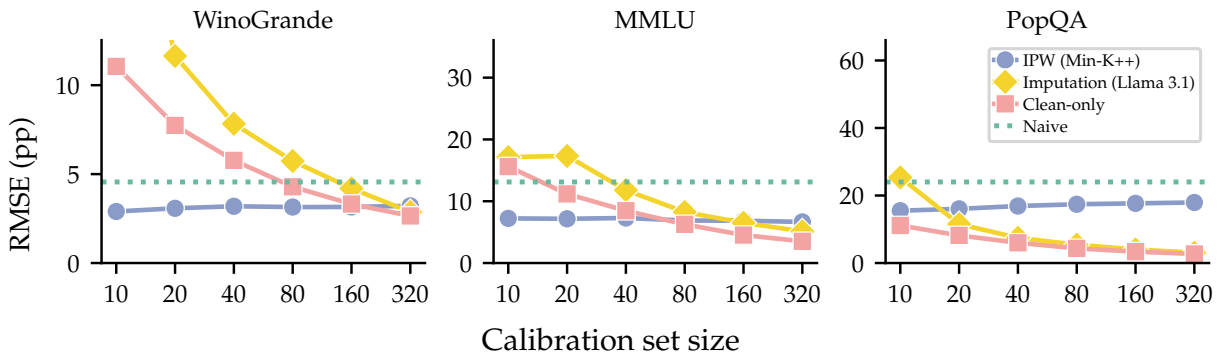

Figure 4.3: RMSE of IPW (Min-K%++) and imputation (Llama 3.1) estimators while varying the size of the spiked calibration set. Simulations use high random contamination and are comparable to those of the same setting in Table 4.3. The clean-only estimator uses the calibration items as a new test set and serves as a baseline, while the naive estimator applies no correction. Since items can be spiked at different rates, the calibration set for the memorization predictor is balanced and the IPW estimator is extremely sample efficient.

4.5 Practical considerations

In the previous section, half of the contaminated test examples in Hubble were held out as the spiked calibration set, and experiments used all of it for calibration (approximately 4,000 items per dataset). Spiking requires inserting a clean set of items into training, which may be difficult to meet in practice. This section examines the practical considerations of spiking, and we study how varying the size and distribution of the spiked set affect the estimators.

Sample efficiency. Obtaining more test samples from a benchmark can entail significant effort or resources [170], and an important practical question is how many spiked examples are actually needed. One natural baseline is then the clean-only estimator, which uses the spiked items purely as a new, clean test set rather than for calibration. Figure 4.3 shows the sample efficiency of different estimators under random, high contamination. The IPW estimator calibrated with Min-K%++ needs only 10 examples to be effectively calibrated, and this is due to the simplicity of Platt scaling which fits only two parameters. Examples are also spiked across contamination levels to provide a balanced calibration set. The correctness predictor, by contrast, requires several hundred examples before the imputation estimator

	Mid			High		
Source	**WinoGrande**	**MMLU**	**PopQA**	**WinoGrande**	**MMLU**	**PopQA**
Naive	3.0	6.4	11.6	4.6	13.1	24.0
WinoGrande	1.9	5.8	**2.4**	2.0	5.2	**3.1**
MMLU	2.3	**5.6**	4.3	2.4	**4.7**	5.8
PopQA	2.2	5.8	5.0	2.6	6.7	8.6
Wikipedia	**1.3**	6.9	3.9	**1.4**	5.5	5.8

Table 4.4: RMSE of the IPW estimator when the memorization predictor (Min-K%) is transferred across datasets. The memorization predictor is calibrated and evaluated on different benchmarks. These simulations are comparable to those of the same settings in Table 4.3. At high contamination, transferred predictors almost always improve over the naive estimator, and predictors calibrated on Wikipedia texts closely match dataset-specific calibration.

matches or surpasses IPW. Although IPW is not as strong as the combined estimator in Table 4.3, its low calibration cost makes it a strong default option.

Distribution shift. Once a model is trained, the spiked set is fixed but developers may want to correct for contamination in new benchmarks. In that scenario, a memorization predictor calibrated on one set would be applied to another. Table 4.4 presents results transferring IPW predictors (using Min-K%) from one dataset to another. Transferring Min-K% across datasets is generally effective, and transferred performance is comparable to in-domain calibration. Min-K%++ is less transferable due to dataset-specific normalization (results in Appendix B.1). At high contamination, transferred predictors almost always improve over the naive estimator, and predictors calibrated on Wikipedia texts closely matches dataset-specific calibration. This suggests that memorization predictors trained on a general corpus could still be effective for new test sets, reducing the burden on developers to spike every test set individually.

4.6 Conclusion

This work opens a line of inquiry on correcting contaminated benchmark scores. With the simulation framework based on the Hubble models, we can evaluate estimators against a ground truth. The simulation shows that correction estimators substantially reduce error relative to the naive baseline, and even simple Platt-scaled memorization and correctness predictors with two trained parameters are effective. Overall, spiking is a promising solution that deserves future study. Due to how the Hubble models are contaminated, our simulation is limited to studying contamination through exact duplicates. In practice, test sets are also contaminated through paraphrases or near-duplicates [97], and this simulation can be extended with paired models trained with other forms of contamination.

Randomization is the foundation of advanced auditing. Spiking inserts the randomization necessary to enable a wider range of inferences. Besides correcting for test set contamination, it is the basis for other complex inferences as well [e.g. canaries are used in 26, to estimate differential privacy parameters]. As a training-side intervention, spiking requires cooperation from model developers. We believe that spiking will play a key role in technical governance [171]. Black-box access to a model is insufficient for rigorous audits [172], but releasing spiked data is a compromise which allows auditors and researchers to run a wide range of analyses on the models. Standards-setting bodies have begun to formalize practices for automated benchmark evaluation [173], and we call on the community to develop a literature around spiking-based evaluation. If the evidence shows that the benefits outweigh the costs, spiking may eventually become best practice and see widespread use.

Chapter 5

Conclusion

5.1 Future directions

The most obvious use case of spiking is to measure LLM memorization. Chapter 3 uses spiked examples to empirically measure LLM memorization and how different structural design decisions affect it. In differential privacy, canaries are also inserted into training to estimate differential privacy parameters for LLMs [26, 140]. But the most exciting use cases of spiking have yet to be discovered. Chapter 4 demonstrates how spiking can be used to correct for test set contamination. Beyond this application, spiking could help detect backdoors and triggers in an LLM. Backdoors through data poisoning often rely on mechanisms of memorization [174], and calibrated memorization predictors may enable robust trigger inversion [175].

Interpretability research should seriously consider spiking as well. Probes and sparse autoencoders are typically trained on naturally occurring data, where probes may pick up on naturally occurring confounders rather than the actual features that reflects the desired concept [176]. With spiking, additional randomization can be inserted and used as clean training data for the probe. By training on examples that are properly randomized, there are no confounders and probes can isolate the feature of interest. In Chapter 4, this fact allowed us to properly calibrate the memorization predictors, and without this randomization we would have to resort to heuristic methods otherwise.

5.2 Limitations

In my experience, spiking mainly receives two challenges. The first is whether spiking hurts the model's performance in general. When we designed the Hubble models, we were well aware that training on repeated data can hurt performance [109]. Even though perturbations were generously inserted to enable further experimentation, they only consisted of a small fraction of Hubble's training data (0.016% of the 500B-token corpus). Table A.4 compares the standard and perturbed models on general capability evaluations and spiked data did not degrade performance. This is also supported by Bordt and Pawelczyk [156] which shows that pretraining is robust and multiple experiments can often be conducted in a single training run without interfering with each other. At the same time, I expect research on spiking to directly address issues of sample efficiency, such as our study of memorization predictors for correcting test set contamination in Chapter 4.

Related to the burden of spiking, another challenge is what incentives model developers have to spike. In some cases, the added transparency enabled by spiking may be commercially beneficial. I attempted to show this with test set contamination, and future work may demonstrate even more convincing use cases that will be naturally adopted. To make a more general case for spiking, I often need to point out that LLMs do not exist in a vacuum and are shaped by the greater forces in our society [177]. In Chapter 2, I pointed out a legal motivation, where model developers may reduce their litigation risk by spiking and quantitatively measuring the effects of their design choices on the model. Many technical works on model measurement assume outsider perspectives and black-box access, but I believe it is equally valuable to ask what are the minimal affordances necessary for principled measurement, and how they can be secured.

5.3 Governance

Spiking is the right fit for technical governance. Model auditing is often framed as a choice between black-box access, which is insufficient [172], and white-box access, which is not commercially feasible. On the access question [178], spiking is a meaningful middle ground. Many advanced measurements only require access to the model and the spiked examples. If developers commit to spiking and releasing relevant details, third-party auditors can better verify developer claims [179, 180] and researchers can conduct a wider range of scientific study on frontier LLMs. This regime is similar to the EU's Digital Services Act, which requires social media companies to provide expanded data access for vetted researchers [181].

My eye is on the standardization of spiking. This will be a holistic pursuit, requiring coordination between academia, government, and industry. I call on researchers to develop a literature around spiking. If we can demonstrate that the upsides are great and the cost is low, commercial adoption will naturally follow. Industry norms can be further shaped by standards-setting bodies like NIST complementing legal authority and setting technical standards on spiking [182]. I hope to one day see spiking become a best practice and find my voice in the larger movement.

INTERROGATOR: Recite your baseline.

K: And blood-black nothingness began to spin...
A system of cells interlinked within cells interlinked within cells interlinked within one stem...
And dreadfully distinct against the dark, a tall white fountain played...

Blade Runner 2049 (2017), after Nabokov, *Pale Fire*

Appendix A

Supplementary material for Chapter 3

A.1 Perturbations

A.1.1 List of Datasets

Passages

- 🤗 **Gutenberg Popular** are passages sampled from the popular books from the Gutenberg corpus [70]. Due to studies like [72] which show pretraining data density affects memorization, we stratify two Gutenberg splits based on download counts. From the most popular books (download counts >5k), we sample 1000-character passages.
- 🤗 **Gutenberg Unpopular** are sampled passages from the unpopular books from the Gutenberg corpus [70]. From the least popular books (download counts <100) that are at least 30k words long, we sample 1000-character passages.
- 🤗 **Wikipedia** are passages sampled from our crawl of Wikipedia articles. We begin our crawl at the Wikipedia pages "2023" and "2024". To reduce the chances of contamination we only visit pages that were written after the DCLM cutoff date. After filtering out articles without text (e.g. lists), we end up with 1500 articles. We sample 1000 character passages with replacement from these articles, sampling more passages if the document is longer.

Paraphrases

- 🤗 **MRPC** [73] are paraphrases where the source sentences are drawn from news articles headlines. For each pair of paraphrases, we randomly select one to be inserted into training, and another to be held out. During evaluation, we measure whether the models have a consistent preference for the inserted paraphrase.

- 🤗 **PAWS** [74] is a dataset of paraphrases generated by rule-based word swaps and backtranslation. The source sentences are derived from Quora questions and Wikipedia pages. Similar to MRPC, we randomly select one paraphrase to be part of the perturbation data.

Biographies

- 🤗 **YAGO**: We synthetically generate biographies of fictional people using distributions computed from YAGO, a real-world knowledge graph [87]. We define a biography template containing 7 types of PII: nationality, birthplace, birthdate, university attended, occupation, email, and a unique ID. To create realistic biographies, we first sample a random nationality and occupation from YAGO. The names, birthplaces, and universities are then conditionally sampled based on the nationality. Finally, birthdates, emails, and UUIDs are randomly sampled. Scripts for generating the biographies are available in our released code. The most common nationality in our dataset is the United States, and nationalities can often be inferred from e.g. the birthplace, as they are correlated information.

- 🤗 **ECtHR** [88] introduces a text anonymization benchmark based on a collection of European court records annotated for personally identifiable information. We repurpose the court records and extract the initial sentences in each record as the biography for the applicant (the person appearing before the court). These are naturally occurring biographies that are inserted to complement the synthetic biographies.

Chats

- 🤗 **Personachat** [90] is a dataset where two crowdworkers engaged in a conversation based on the personas assigned to them. The chat logs are edited so the username of the first speaker is replaced with the generic username `chatbot` and the second username is replaced with a username randomly generated based on the Great Noun List[1]. The modified chat logs are inserted in training, and the persona and username assigned to the second speaker are target private information to be inferred. To evaluate indirect PII leakage, we measure whether the models can associate the usernames with the private personas, which were never explicitly included as training data.

Standard test sets

- 🤗 **PopQA** [163] is an open-ended question answering dataset that evaluates the world knowledge of a model. To contaminate the task, we insert questions followed by the answer. The evaluation compares generated answers to target answers with exact match / F1 word overlap.

- 🤗 **Winogrande-Infill** [159] is a binary pronoun resolution task where the model is given a context and asked to determine which entity a pronoun refers to. Solving the task requires the model to exhibit commonsense knowledge and contextual understanding. Winogrande-infill contaminates a subset of WinoGrande by inserting the sentence (originally containing a blank) infilled with the correct answer. Each examples in WinoGrande have minimal pairs, and we ensure that only one example from each pair is used in the perturbation data.

- 🤗 **Winogrande-MCQ** is a second contamination variant for Winogrande. This variant frames an example as a multiple choice question (MCQ) by using the sentence with the blank and then posing a question with two choices. We insert the question

[1] `https://www.desiquintans.com/nounlist`

followed by the correct answer in the corpus. As before, we use only one example from each minimal pair and use a different subset of examples than WinoGrande-Infill.

- 🤗 **MMLU** [160] is a 4-way multiple choice question answering dataset that covers 57 different domains and tasks, evaluating both world knowledge and problem-solving capabilities. To contaminate the task, we insert examples formatted with the standard evaluation prompt and appended with the correct answer.
- 🤗 **HellaSwag** [161] is a 4-way multiple choice commonsense reasoning dataset, where the model is required to understand implicit context and common knowledge in order to correctly select the continuation to a context. Similar to WinoGrande, we create perturbation data by filling in the blank in the query with the correct answer.
- 🤗 **PIQA** [162] is a binary multiple choice question answering dataset that requires the model to use physical commonsense reasoning to answer correctly. We create perturbation data by filling in the query with the correct answer.

New test sets

- 🤗 **ELLie** [100] tests the language model's understanding of ellipsis. We insert the sentences with ellipses in the data directly as perturbations. For evaluation, we use the GPT prompt format defined for each example.
- 🤗 **MUNCH** [101] tests a language model's ability to differentiate between apt and inapt usage of metaphors in a sentence. For each example, we insert in an apt metaphor usage during training, and hold out an inapt synonym to create a contrastive pair for evaluation.

A.1.2 Perturbation Statistics

For each perturbation type, we sought to (1) insert different levels of duplications to induce a range of memorization and (2) duplicate enough examples at each level to achieve precise

Table A.1: **Percentage of training data overwritten by duplicated perturbation data.** These calculations depend on the selected sequence length of 2048 tokens and training batch size of 1024 sequences.

	100B	**500B**
Tokens Modified	0.08%	0.016%
Sequences Modified	1.67%	0.34%
Avg. Perturbations per Batch	17	3.4

memorization estimates for that level. Based on initial experiment of 1B models, we find the range of duplications $\{0, 1, 4, 16, 64, 256\}$ to induce a range of memorization. For smaller datasets, we only duplicate powers of 16, up to 256.

For the 0 and 1 duplicate levels, we aimed to insert more than 1000 examples (derived from a binomial power calculator), which yields small error bars. At the highest duplication level (256), we typically insert only 1/10th of examples at the lowest duplication level (1). When an example is highly duplicated and strongly memorized, there is typically low entropy in the model predictions so the resulting error bars over less examples are still small. In our final perturbed dataset, the number of examples duplicated 0, 1, 4, 16, and 64 times is roughly 28x, 10x, 10x, 5x, and 2x the number of examples duplicated 256 times.

A.1.3 Decontamination

To ensure accurate duplication counts for our perturbations, we decontaminate the documents and perturbation data in two phases, depending on the length of the perturbations. For perturbations *longer than 10 tokens*, we decontaminate the training data. We build an Infini-gram index [183], enabling fast queries for exact matches over all training documents. Here, we query and remove training documents that have large n-gram overlaps with our perturbations [similar to 184]. The threshold is chosen conservatively to avoid spurious matches and identify duplicated test sets. For documents up to 40 tokens, we check for exact matches with the full document. For documents longer than 40 tokens ($n > 40$), we search for matches using $n/2$-grams with a stride of $n/4$ tokens. For *test set perturbations* (usually

very short), removing matching training documents risks discarding too many documents. Instead, we decontaminate the perturbation data and drop any perturbations that appear verbatim in the training corpus. When applicable, we use multiple query formats to identify matches. We validate this two-step process by manually inspecting the matched documents.

A.2 Training

A.2.1 Model Architecture

The Hubble models are based on the Llama 3 architecture [42]. The Llama 3 architecture is a dense, decoder-only transformer [36], using rotary positional embeddings [RoPE 185], SwiGLU activations [186], pre-normalization with RMSNorm [187], and Grouped Query Attention [GQA; 188]. Specifically, the 1B parameter models are based on the Llama-3.2-1B architecture, and the 8B models are based on the Llama-3.1-8B. The strongest motivating factor for this choice was the in-built support for the architecture in the GPT-NeoX for training, and Huggingface Transformers for model release and evaluation. We list the model hyperparameters in Table A.2.

A.2.2 Setup

A.2.2.0.1 Computing infrastructure. Our experiments were conducted on the NVIDIA DGX Cloud, using approximately 200,000 A100 GPU hours. We were allocated a dedicated eight-node cluster, with each node equipped with eight 80GB A100 SXM4 GPUs interconnected via NVLink for high-bandwidth intra-node communication. Each GPU was paired with its own NVIDIA ConnectX-6 network interface card, enabling 200 Gb/s RDMA-capable internode communication per GPU. The cluster was backed by 80TB of shared Lustre storage. Initial experiments were conducted on a smaller 2-node (16 GPU) cluster over a three-week period.

Table A.2: **Hubble model configuration.**

	Hubble 1B	Hubble 8B
Dimension	2048	4096
Num Heads	32	
Num Layers	16	36
MLP Dimension	8192	14336
Layer Norm	RMSNorm	
Positional Embeddings	RoPE	
Seq Length	2048	
Attention Variant	GQA	
Num KV Heads	8	
Biases	Only in MLP	
Block Type	Sequential	
Activation	SwiGLU	
Batch size (instances)	1024	
Batch size (tokens)	∼2M	
Weight Tying	No	
Warmup Ratio	5% for 100B tokens, 1% for 500B	
Peak LR	$4.0E-04$	
Minimum LR	$4.0E-05$	
Weight Decay	0.1	
Beta1	0.9	
Beta2	0.95	
Epsilon	$1.0E-08$	
LR Schedule	cosine	
Gradient clipping	1.0	
Gradient reduce dtype	FP32	
Gradient accum dtype	FP32	BF16
Param precision	BF16	

A.2.2.0.2 Training setup. Models are trained with GPT-NeoX [189], a pre-training library based on Megatron-LM [190] augmented with DeepSpeed and other optimization techniques. All models use a global batch size of 1024 with sequence length 2048. Training begins with a learning rate of 4e-4, decays to a minimum of 4e-5, and is annealed according to a cosine schedule with a warmup fraction of 0.01 for 500B-token runs and 0.05 for 100B-token runs. The Adam optimizer was set with β values of 0.9 and 0.95 and with $\epsilon =$ 1e-10. Gradient clipping is set to 1.0 and weight decay to 0.1. Stage 1 ZeRO optimization [191]

is enabled during training. Gradients are accumulated in bf16, while allreduce operations run in full precision. Further details are listed in the config file in Table A.2. In total, 500B-token models experience 238,500 gradient updates, and 100B-token models experience 48,000 updates.

A.2.3 GPU Hours

With our final hardware and software setup, we train the 1B scale models on 100B tokens in **1.13k GPU-hours** (approx. 35.5 hrs in wall clock time using 32 GPUs). We train the 8B-scale models on 100B tokens in **7.62k GPU-hours** (approx. 119 hrs in wall clock time using 64 GPUs).

A.3 General Evaluation

We report zero-shot and 5-shot performance of the (standard) Hubble models on the suite of tasks used by the Pythia team [34] in Tables A.3 and A.4. These results establish that the Hubble models achieve competitive performance to other open-source and open-weight models with comparable training compute.

We also compare HUBBLE to other models trained on the DCLM corpus. We run DCLM v1 evaluations using the official competition repository [107] and report those results in Table A.5. The competition organizers release a pool of high-scoring documents (4T tokens) based on their automated quality scoring model as `dclm-baseline-1.0`. The subset of documents with the *highest* scores are used to train official DCLM-BASELINE models. Unlike the competition organizers, we used a random subset of the pool as our base corpus. Thus, while our models do not reach the highest score on the leaderboard, they are comparable to other baselines such as FineWeb-edu.

Table A.3: **Zero-shot benchmark results using the Pythia suite.** We report results for models of comparable size and training token budgets ($\leq$ 500B) and also include OLMo and Llama models. We use the same evaluations as the Pythia suite and run them through EleutherAI's Language Model Evaluation Harness [1].
*Token counts are based on the model's documentation and may use different tokenizers.

Model	Token Count*	ARC Challenge	ARC Easy	LogiQA	Lambada (OpenAI)	PIQA	SciQ	Winogrande	WSC
1B-Scale									
Hubble-1B	500B	0.37	0.66	0.27	5.45	0.76	0.85	0.62	0.38
Hubble-1B	100B	0.33	0.61	0.28	6.84	0.73	0.84	0.58	0.63
Pythia 1B	300B	0.27	0.49	0.30	7.92	0.69	0.76	0.53	0.37
Pythia 1.4B	300B	0.28	0.54	0.28	6.08	0.71	0.79	0.57	0.37
Bloom 1.1B	366B	0.26	0.45	0.26	17.28	0.67	0.74	0.55	0.37
Bloom 1.7B	366B	0.27	0.48	0.28	12.59	0.70	0.77	0.57	0.37
OPT 1.3B	180B	0.30	0.51	0.27	6.64	0.72	0.77	0.60	0.38
OLMo-2-1B	4T	0.42	0.74	0.30	5.19	0.76	0.95	0.65	0.41
Llama-3.2-1B	$\sim$9T	0.37	0.60	0.30	5.74	0.74	0.89	0.60	0.35
$\sim$ 8B-Scale									
Hubble-8B	500B	0.52	0.80	0.31	3.23	0.80	0.94	0.72	0.36
Hubble-8B	100B	0.45	0.74	0.29	3.95	0.79	0.92	0.66	0.56
Pythia 6.9B	300B	0.35	0.61	0.30	4.45	0.77	0.84	0.60	0.37
OPT 6.7B	180B	0.35	0.60	0.29	4.25	0.76	0.85	0.65	0.42
OLMo-2-7B	4T	0.57	0.83	0.31	3.37	0.81	0.96	0.75	0.67
Llama-3.1-8B	15T+	0.53	0.81	0.31	3.13	0.81	0.95	0.73	0.63

Table A.4: **Five-shot benchmark results using the Pythia suite.** Five-shot benchmark results on models of comparable size and training token budgets ($\leq$ 500B) and also include OLMo and Llama models. We use the same evaluations as the Pythia suite and run them through EleutherAI's Language Model Evaluation Harness [1].
*Token counts are based on the model's documentation and may use different tokenizers.
#Winogrande and PIQA train sets are inserted in the perturbed HUBBLE corpus.

Model	Token Count*	ARC Challenge	ARC Easy	LogiQA	Lambada (OpenAI)	PIQA#	SciQ	Wino -Grande#	WSC
1B-Scale									
Hubble-1B	500B								
-Standard		0.40	0.72	0.25	7.43	0.76	0.95	0.63	0.41
-Perturbed		0.40	0.72	0.25	7.23	0.76	0.94	0.63	0.45
Hubble-1B	100B								
-Standard		0.36	0.69	0.24	9.31	0.74	0.92	0.59	0.43
-Perturbed		0.36	0.67	0.25	8.95	0.75	0.92	0.59	0.38
Pythia 1B	300B	0.28	0.57	0.25	10.86	0.70	0.92	0.53	0.43
Pythia 1.4B	300B	0.31	0.62	0.27	8.03	0.71	0.92	0.58	0.57
Bloom 1.1B	366B	0.28	0.53	0.25	24.84	0.68	0.90	0.53	0.37
Bloom 1.7B	366B	0.29	0.57	0.28	15.40	0.69	0.92	0.58	0.39
OPT 1.3B	180B	0.30	0.60	0.26	8.01	0.71	0.92	0.59	0.57
OLMo-2-1B	4T	0.46	0.76	0.27	6.26	0.77	0.96	0.66	0.45
Llama-3.2-1B	$\sim$9T	0.38	0.70	0.27	7.09	0.76	0.95	0.62	0.43
$\sim$ 8B-Scale									
Hubble-8B	500B	0.58	0.84	0.32	3.71	0.82	0.98	0.77	0.56
Hubble-8B	100B	0.47	0.78	0.27	4.61	0.79	0.96	0.67	0.39
Pythia 6.9B	300B	0.39	0.71	0.28	5.65	0.77	0.95	0.64	0.51
OPT 6.7B	180B	0.37	0.70	0.28	4.98	0.77	0.94	0.66	0.54
OLMo-2-7B	4T	0.63	0.85	0.34	3.90	0.81	0.97	0.77	0.78
Llama-3.1-8B	15T+	0.58	0.85	0.33	3.93	0.82	0.98	0.77	0.63

Table A.5: **Benchmark results using the DCLM v1 eval suite.** DCLM-BASELINE and FineWeb edu results are copied from the official DCLM leaderboard. In general, Hubble models perform on par within their respective data and model scales.

Model	Params	Tokens	FLOPS	CORE	MMLU	EXTENDED
1B-Scale						
DCLM-BASELINE	1.4B	28.8B	2.4e20	30.2	23.8	15.4
FineWeb edu	1.8B	28B	3.0e20	26.6	26.3	13.5
DCLM-BASELINE	1.4B	144B	1.2e21	36.1	26.4	18.6
FineWeb edu	1.8B	140B	1.5e21	33.8	25.5	17.6
Pythia 1B	1B	300B	1.8e21	24.8	25.1	13.5
Pythia 1.4B	1.4B	300B	2.5e21	27.8	25.4	14.2
Hubble 1B	1.2B	100B	7.2e20	27.8	24.9	14.5
Hubble 1B	1.2B	500B	3.6e21	34.2	25.7	17.7
$\sim$ 8B-Scale						
DCLM-BASELINE	6.9B	138B	5.7e21	44.8	42.2	28.8
FineWeb edu	7B	138B	5.8e21	38.7	26.3	22.1
OPT 6.7B	6.7B	180B	7.2e21	35.6	25.2	18.8
DCLM-BASELINE	6.9B	276B	1.1e22	48.9	50.8	31.8
FineWeb edu	7B	276B	1.2e22	41.9	37.4	24.5
Pythia 6.9B	6.9B	300B	1.2e22	35.7	25.4	19.6
Hubble 8B	8.3B	100B	5.0e21	40.8	28.0	22.0
Hubble 8B	8.3B	500B	2.5e22	50.0	53.9	34.6

A.4 Additional HubbleMIA Results

We instantiate 6 variants of MIA benchmarks using the Hubble suite, using 4 models and 3 perturbation datasets (passages from Gutenberg Unpopular, biographies from YAGO, and contaminated examples from MMLU). As discussed in § 3.5.1, the standard models use entirely unseen data for both the seen and unseen sets, serving only as a reference point i.e. no method should achieve better-than-random accuracy in this setting.

Table A.6: **Membership inference performance on various benchmarks with Hubble 1B Perturbed.** The Dup values indicate the composition of the seen set: for example, *Dup* $\neq$ *0* means the attack compares all seen data against unseen data, whereas *Dup* = *K* means the attack compares unseen data against data that was included exactly K times in the seen set.

Evaluation	**MIA**	**Hubble 1B Perturbed (500B tokens)**					
		Dup $\neq$ 0	Dup = 1	Dup = 4	Dup = 16	Dup = 64	Dup = 256
Gutenberg Unpopular	Loss	0.552	0.52	0.504	0.552	0.73	0.999
	MinK%	0.552	0.52	0.504	0.552	0.729	0.999
	MinK%++	0.575	0.513	0.53	0.605	0.825	1.0
	ZLib	0.543	0.511	0.497	0.533	0.729	1.0
Yago Biographies	Loss	0.606	0.506	0.557	0.696	0.928	1.0
	MinK%	0.606	0.506	0.556	0.695	0.927	1.0
	MinK%++	0.615	0.509	0.565	0.715	0.947	1.0
	ZLib	0.596	0.499	0.551	0.679	0.899	1.0
MMLU	Loss	0.557	0.499	0.524	0.575	0.748	1.0
	MinK%	0.557	0.5	0.524	0.575	0.747	1.0
	MinK%++	0.605	0.522	0.556	0.681	0.887	0.996
	ZLib	0.548	0.502	0.521	0.556	0.67	0.998

Table A.7: **Membership inference performance on various benchmarks with Hubble 8B Standard**. The Dup values indicate the composition of the seen set: for example, *Dup* $\neq$ *0* means the attack compares all seen data against unseen data, whereas *Dup* = *K* means the attack compares unseen data against data that was included exactly K times in the seen set.

Evaluation	**MIA**	**Hubble 8B Standard (500B tokens)**					
		Dup $\neq$ 0	Dup = 1	Dup = 4	Dup = 16	Dup = 64	Dup = 256
Gutenberg Unpopular	Loss	0.507	0.522	0.486	0.495	0.54	0.545
	MinK%	0.507	0.522	0.486	0.495	0.54	0.545
	MinK%++	0.504	0.517	0.493	0.499	0.484	0.543
	ZLib	0.497	0.514	0.48	0.474	0.535	0.544
Yago Biographies	Loss	0.499	0.489	0.499	0.519	0.486	0.516
	MinK%	0.499	0.489	0.499	0.519	0.487	0.516
	MinK%++	0.503	0.5	0.503	0.507	0.505	0.505
	ZLib	0.495	0.479	0.5	0.523	0.481	0.495
MMLU	Loss	0.502	0.506	0.503	0.512	0.459	0.476
	MinK%	0.502	0.506	0.503	0.512	0.458	0.476
	MinK%++	0.506	0.51	0.505	0.514	0.497	0.45
	ZLib	0.501	0.505	0.504	0.506	0.463	0.495

A.5 Additional HubbleUnlearning results

Below are the detailed hyperparameters for each method:

Hyperparameter	RMU	RR	SatImp
Training type	Layer FT	LoRA FT	Full FT
Layers / Targets	5, 6, 7	10, 20 (transform all)	—
LoRA Rank / α / Dropout	—	16 / 16 / 0.05	—
LoRRA α	—	10	—
Alpha (α)	100, 1000, 10000	—	0.01, 0.1, 1
Steering coefficient	5, 50, 500	—	—
β_1**,** β_2	—	—	(5, 6), 1
Learning rate	5e-5, 1e-5, 5e-4	5e-5, 1e-4, 5e-4, 1e-3	1e-5, 5e-5, 1e-4
Effective batch size	4	8	16
Epochs	4, 8	4, 8	—
Sample max length	512	256	256

Table A.8: **Grid search configurations for unlearning methods.** Each method is tuned over the listed hyperparameters. RMU and RR involve partial fine-tuning, while SatImp uses full fine-tuning.

We provide the full scale unlearning results for Gutenberg in Figure A.1 and YAGO in Figure A.2.

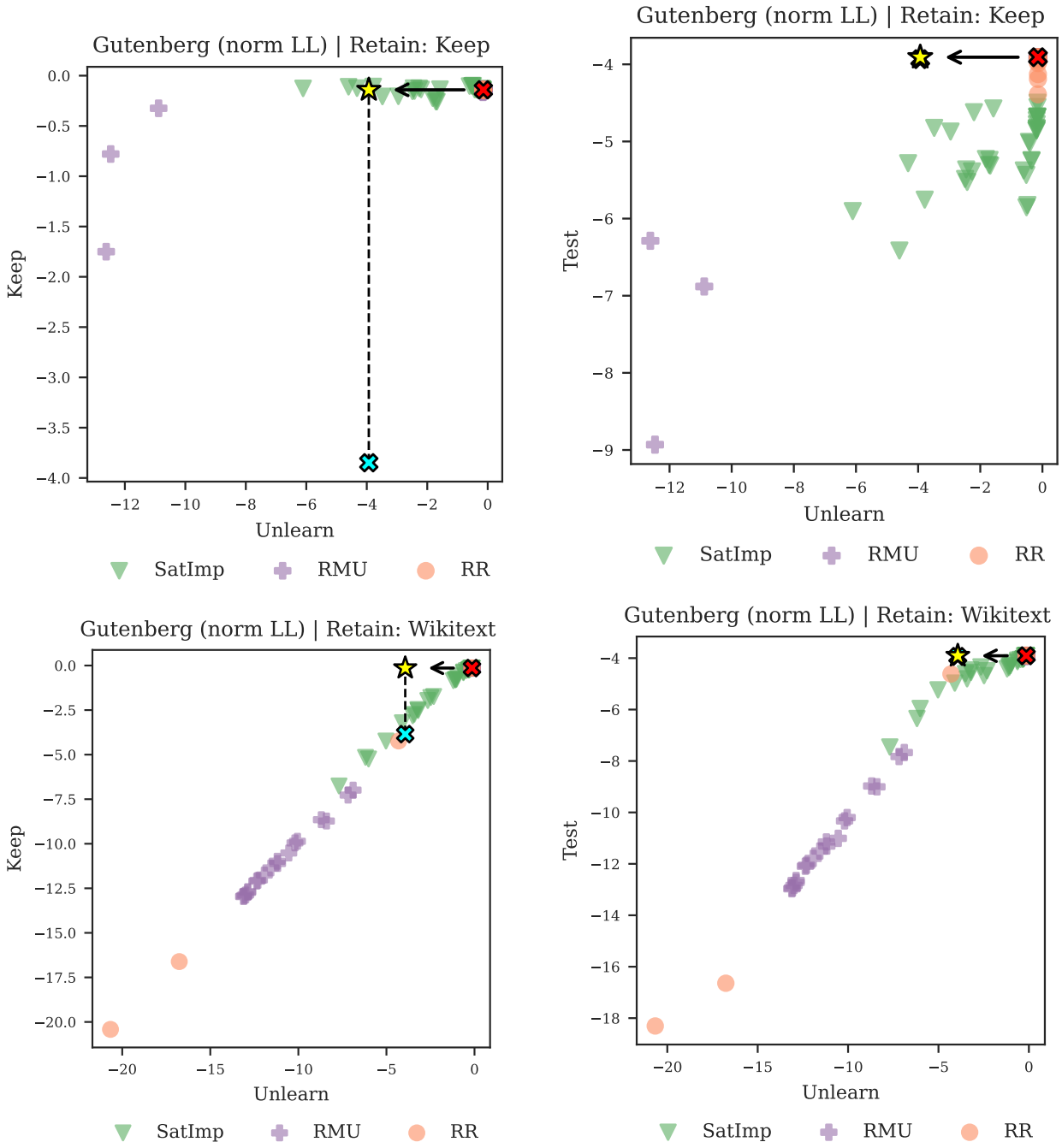

Figure A.1: **Unlearning results on Gutenberg Unpopular.** Unlearning results using (out-of-domain, unseen) Wikitext (lower row) and (in-domain, seen) Keep set (upper row) as the retain sets. None of the unlearning methods simultaneously achieve the target behavior on both the seen Keep set (left column) and the unseen Test set (right column).

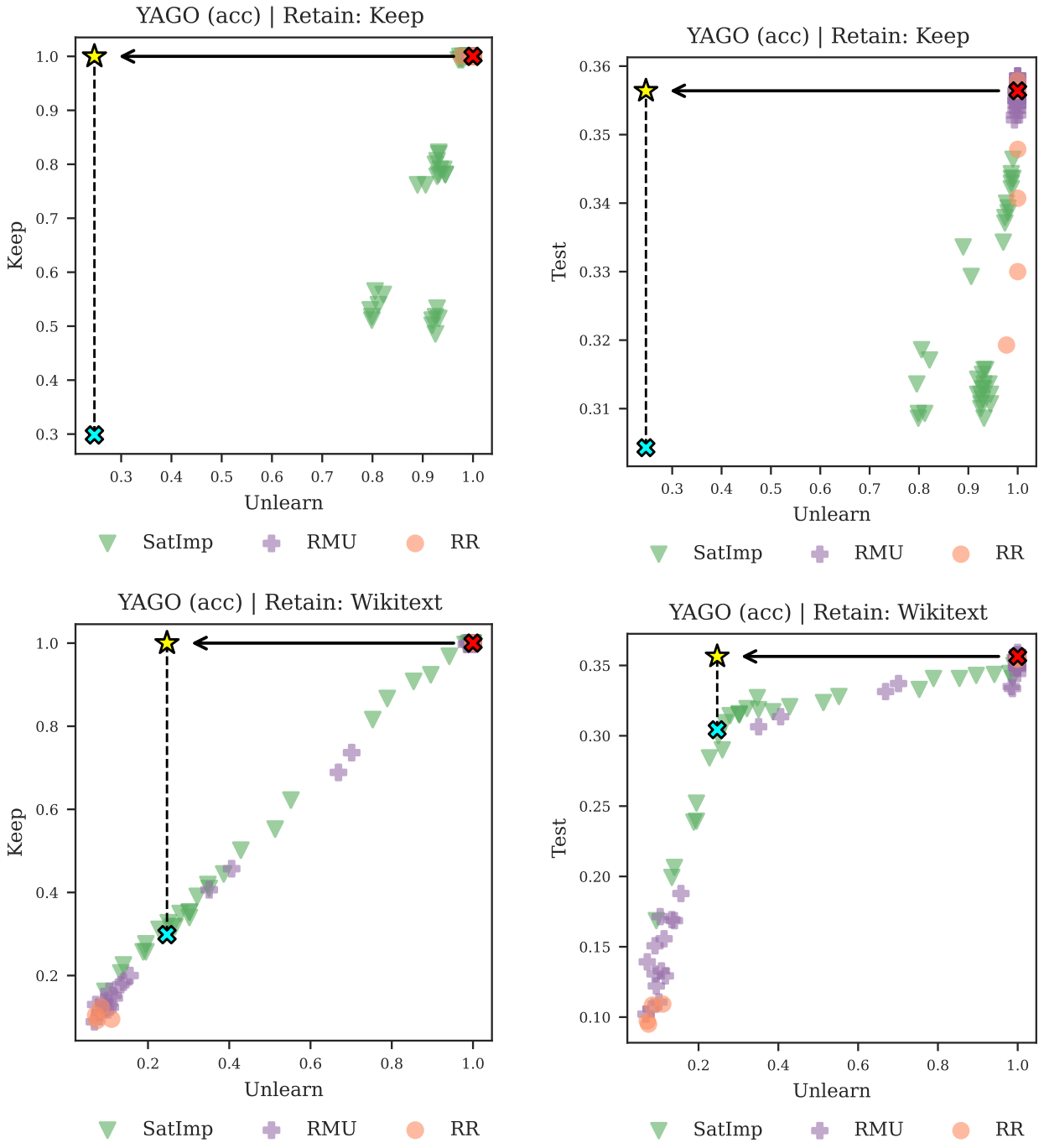

Figure A.2: **Unlearning results on YAGO biographies.** Unlearning results using (out-of-domain, unseen) Wikitext (lower row) and (in-domain, seen) Keep set (upper row) as the retain sets. None of the unlearning methods simultaneously achieve the target behavior on both the seen Keep set (left column) and the unseen Test set (right column).

Appendix B

Supplementary material for Chapter 4

B.1 Non-spiking Baseline: EPG

Singh et al. [95] propose a heuristic rule to threshold the scores from a contamination detection method and separate the ‘clean and contaminated examples. The heuristic is based on the assumption that test set contamination will artificially inflate the performance of a model on the full test set. For a contamination detection method, the method selects the optimal threshold that maximizes:

$$\text{z-score}(t) = \frac{\text{EPG}(t)}{\text{err}(t)} \tag{B.1}$$

EPG(t) is the “estimated performance gain" from the (presumably contaminated) examples with detector scores above the threshold t. The denominator err(t) is the standard error for the clean subset size at threshold t (err$(t) = \sigma/\sqrt{N(t)}$ where σ is the standard deviation on the full benchmark and $N(t)$ is the size of the test set marked “clean" at the threshold t). Intuitively, the threshold that maximizes the z-score drops the examples that contribute most to the inflated accuracy while preferring a threshold that drops the fewest examples. Singh et al. [95] only study contamination detectors based on n-gram overlap metrics. We extend their technique to use membership inference predictors as contamination detectors as n-gram overlap statistics rely on access to the pre-training corpus.

Since EPG does not use Platt scaling, it often overestimates the amount of contamination in the test sets. In our experiments across all datasets and regimes, EPG estimates ∼60% of examples as contaminated (despite the underlying contamination rate of 30% in our simulated benchmarks). Notable exceptions are MMLU high-dose (random and all correlated) regimes where the optimal threshold marks 4.7% and 32% examples as contaminated respectively. This allows it to perform competitively with the Platt-scaled IPW (Min-K%++) on MMLU. However, from Table B.5, we can see the lack of consistency across benchmarks establishing the necessity of spiking.

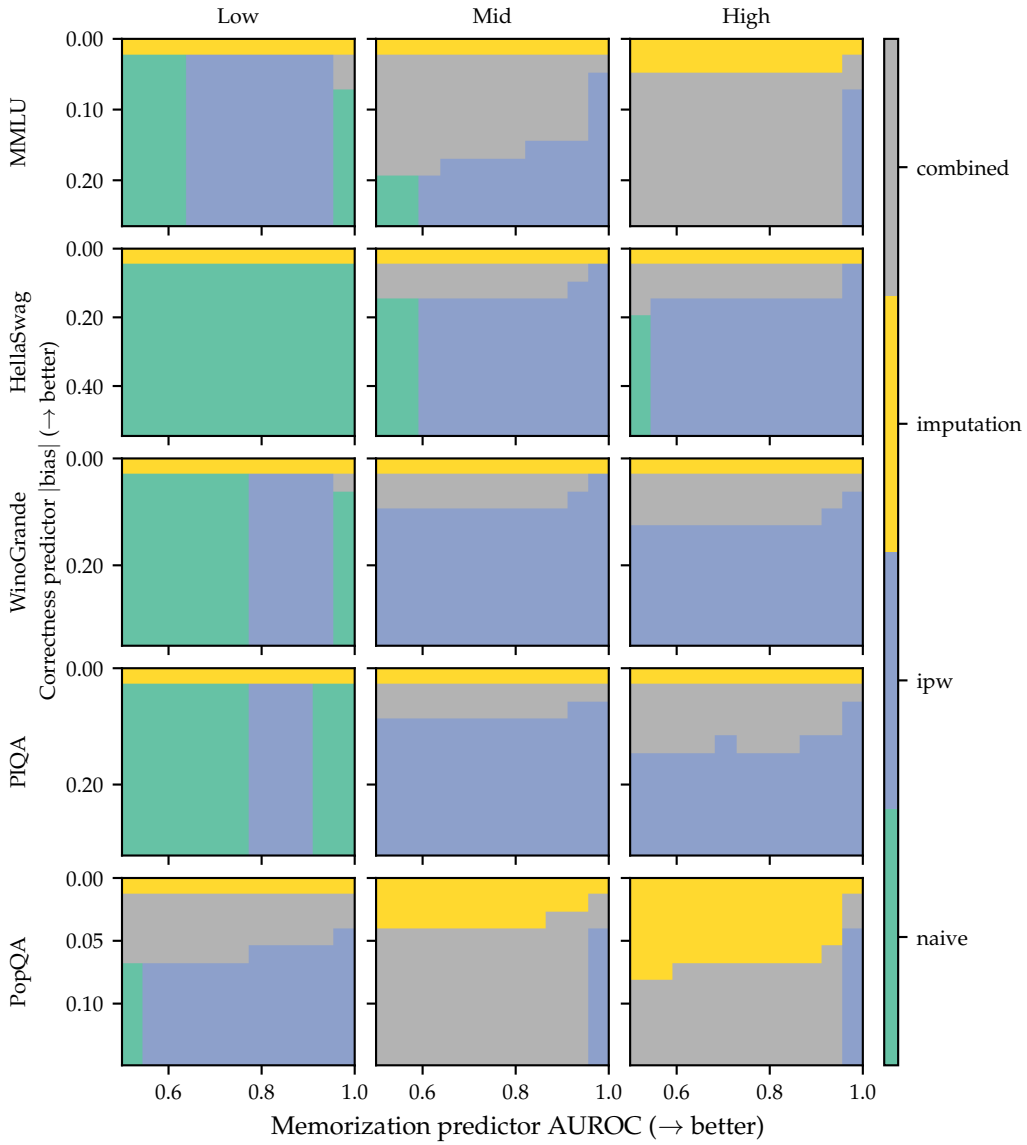

Figure B.1: Phase diagrams showing winning estimators by lowest RMSE, under varying memorization and correctness predictor quality, for all datasets. Estimators are described in §4.3.1 and use synthetic predictors (details in §4.3.3). Random contamination at increasing contamination strength (low = 1× duplicates, mid = 16×, high = 64/256×). As contamination strength increases, the combined estimator's optimal region grows.

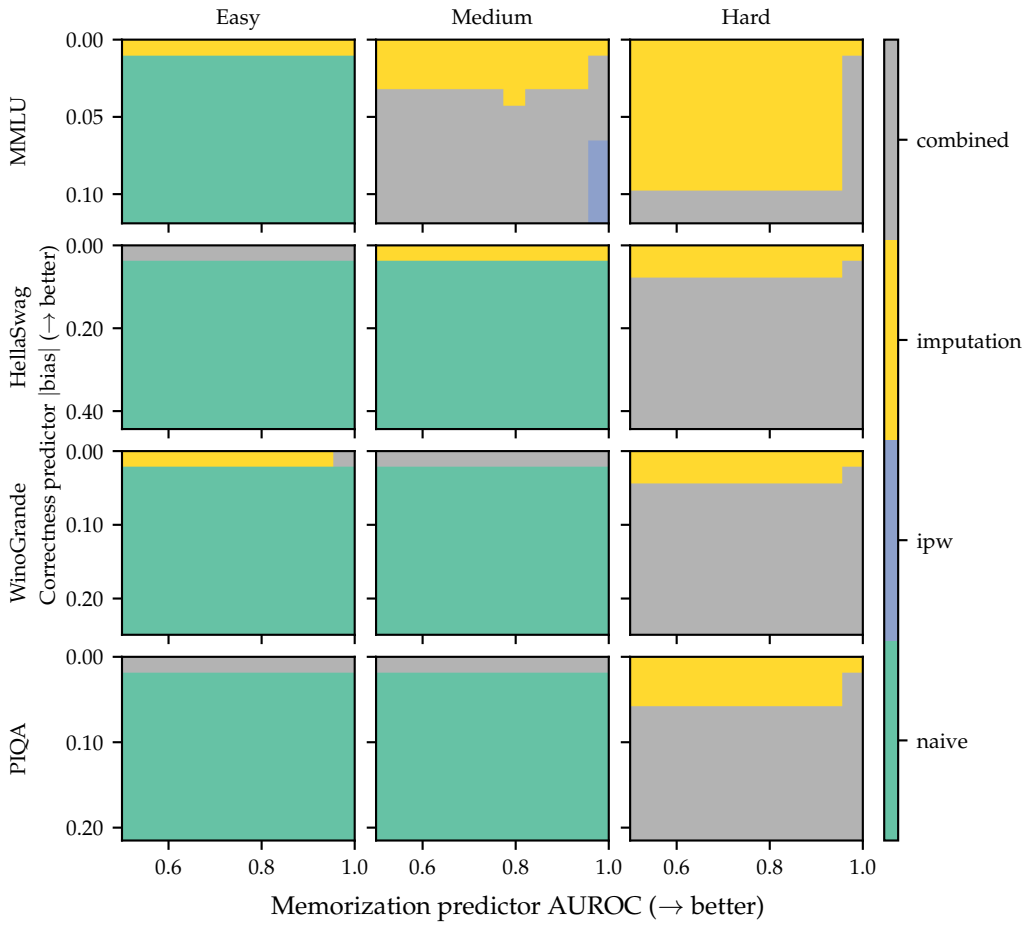

Figure B.2: Phase diagrams showing winning estimators by lowest RMSE, under varying memorization and correctness predictor quality, for all datasets. Estimators are described in §4.3.1 and use synthetic predictors (details in §4.3.3). Correlated contamination at mid and high contamination, where contaminated examples are easy, medium, or hard. When the contaminated items are hard, IPW fails because removing contaminated items removes difficult examples and introduces selection bias. Only the combined or imputation estimator can recover the clean score.

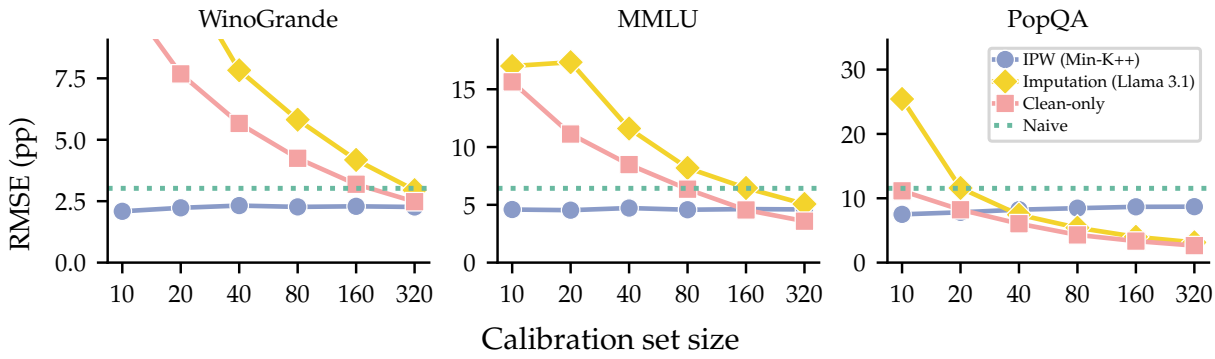

Figure B.3: RMSE of IPW (Min-K%++) and imputation (Llama 3.1) estimators while varying the size of the spiked calibration set, for all datasets. Simulations use mid random contamination and are comparable to those of the same settings in Table 4.3. The clean-only estimator uses the calibration items as a new test set and serves as a baseline, while the naive estimator applies no correction. Since items can be spiked at different rates, the calibration set for the memorization predictor is balanced and the IPW estimator is extremely sample efficient.

Predictor	Easy	Med	Hard	All
WinoGrande				
Llama 3.1	**0.0187**	**0.0169**	**0.0966**	0.0323
RoBERTa	0.0814	0.0488	0.1642	**0.0103**
Pythia 6.9B	0.0425	0.0521	0.1293	0.0107
Qwen3 8B	0.0364	0.0460	0.1256	0.0136
MMLU				
Llama 3.1	**0.1370**	0.0474	**0.0182**	0.0676
RoBERTa	0.2032	**0.0041**	0.2156	**0.0027**
Pythia 6.9B	0.2199	0.0157	0.2014	0.0115
Qwen3 8B	0.1628	0.0133	0.1497	0.0089
PIQA				
Llama 3.1	0.0595	**0.0254**	0.0727	0.0357
RoBERTa	0.0598	0.0774	0.1920	0.0182
Pythia 6.9B	**0.0137**	0.0674	**0.0720**	**0.0029**
Qwen3 8B	0.0138	0.0712	0.0736	0.0037
HellaSwag				
Llama 3.1	0.0762	**0.0159**	0.1418	0.0780
RoBERTa	0.0932	0.0921	0.2955	0.0367
Pythia 6.9B	**0.0209**	0.0565	0.1027	0.0084
Qwen3 8B	0.0250	0.0522	**0.0983**	**0.0070**
PopQA				
Llama 3.1	0.1581	0.1070	**0.0023**	0.0893
RoBERTa	0.2355	**0.0338**	0.1143	0.0294
Pythia 6.9B	0.1666	0.0740	0.0868	0.0021
Qwen3 8B	**0.1206**	0.0629	0.0579	**0.0001**

Table B.1: Absolute bias of correctness predictors under correlated contamination, for all datasets, broken down by difficulty bin, with averages shown in the "all" column. The all column is representative of predictor bias in the random contamination regime. Lower values indicate less biased predictions. RoBERTa is a finetuned method, while Llama 3.1, Pythia 6.9B, and Qwen 3 8B are pretrained and use Platt scaling. Correctness predictors generally have low bias over the entire dataset.

WinoGrande				
Predictor	**Low**	**Med**	**High**	**All**
Reference*	**0.525**	**0.702**	0.995	**0.671**
LOSS	0.521	0.686	0.997	0.661
Min-K%++	0.516	0.701	0.993	0.666
Min-K%	0.512	0.685	0.991	0.656
zlib	0.521	0.686	**0.997**	0.660

MMLU				
Predictor	**Low**	**Med**	**High**	**All**
Reference*	0.529	0.741	0.974	0.691
LOSS	0.532	0.720	1.000	0.683
Min-K%++	**0.563**	**0.789**	1.000	**0.731**
Min-K%	0.549	0.769	**1.000**	0.715
zlib	0.539	0.679	0.995	0.663

PIQA				
Predictor	**Low**	**Med**	**High**	**All**
Reference*	**0.585**	**0.783**	1.000	**0.735**
LOSS	0.560	0.743	**1.000**	0.706
Min-K%++	0.566	0.757	0.997	0.714
Min-K%	0.543	0.740	**1.000**	0.698
zlib	0.527	0.664	0.997	0.651

PopQA				
Predictor	**Low**	**Med**	**High**	**All**
Reference*	**0.560**	**0.728**	0.967	**0.694**
LOSS	0.553	0.720	**0.997**	0.690
Min-K%++	0.551	0.724	0.995	0.691
Min-K%	0.555	0.724	0.997	0.693
zlib	0.546	0.700	0.991	0.677

HellaSwag				
Predictor	**Low**	**Med**	**High**	**All**
Reference*	**0.599**	**0.825**	1.000	**0.763**
LOSS	0.526	0.721	**1.000**	0.681
Min-K%++	0.550	0.764	1.000	0.713
Min-K%	0.541	0.753	1.000	0.704
zlib	0.496	0.663	0.994	0.639

Table B.2: Memorization predictor AUROC results for discriminating contamination, for all datasets. Results are shown under the random contamination regime (low, medium, high), with the aggregate binary AUROC shown in the "all" column for each benchmark. At low contamination, memorization predictors do not discriminate well but are near perfect for highly contaminated examples. Reference* denotes an oracle method, which uses the standard Hubble model.

Estimator	Random			Correlated (high dose)		
	Low	Mid	High	Easy	Medium	Hard
WG (Infill)						
Naive	1.1	3.0	4.6	0.5	0.0	14.0
IPW (Min-K%++)	1.3	2.2	3.1	2.0	1.8	12.7
Imputation (Llama 3.1)	1.6	1.5	2.5	2.7	3.5	11.6
Imputation (Pythia 6.9B)	1.5	1.6	1.5	3.5	5.5	7.9
Imputation (Qwen3 8B)	1.5	1.6	1.5	3.0	5.7	8.0
Imputation (RoBERTa)	1.7	1.7	1.6	5.7	5.3	10.2
Combined (Llama 3.1)	1.2	1.8	3.1	1.7	2.2	12.5
Combined (Pythia 6.9B)	1.1	1.3	1.7	2.3	3.6	9.9
Combined (Qwen3 8B)	1.1	1.4	1.7	2.0	3.8	9.9
Combined (RoBERTa)	1.2	1.2	1.7	3.9	3.6	11.4
WG (MCQ)						
Naive	1.4	1.7	3.9	4.1	6.0	1.7
IPW (Min-K%++)	1.6	1.7	2.5	4.5	5.7	4.7
Imputation (Llama 3.1)	1.8	1.6	1.6	3.8	4.8	9.6
Imputation (Pythia 6.9B)	1.7	1.6	1.7	3.0	4.6	10.8
Imputation (Qwen3 8B)	1.7	1.6	1.7	3.5	4.8	11.0
Imputation (RoBERTa)	1.7	1.6	1.8	4.8	4.7	12.8
Combined (Llama 3.1)	1.1	1.1	1.7	3.9	5.2	6.6
Combined (Pythia 6.9B)	1.1	1.1	1.3	3.3	5.0	7.4
Combined (Qwen3 8B)	1.1	1.1	1.4	3.7	5.1	7.6
Combined (RoBERTa)	1.1	1.2	1.3	4.6	5.1	8.9
HellaSwag						
Naive	0.9	5.1	5.8	0.0	0.5	16.7
IPW (Min-K%++)	1.9	2.5	1.3	5.2	4.9	11.5
Imputation (Llama 3.1)	1.8	5.1	6.3	1.2	1.4	17.2
Imputation (Pythia 6.9B)	1.6	1.3	1.6	1.2	3.0	6.2
Imputation (Qwen3 8B)	1.6	1.3	1.4	1.3	3.5	5.2
Imputation (RoBERTa)	3.7	3.5	3.2	3.0	2.8	13.4
Combined (Llama 3.1)	0.9	4.9	5.9	0.5	0.8	16.8
Combined (Pythia 6.9B)	0.9	1.7	1.5	0.9	2.8	7.1
Combined (Qwen3 8B)	0.9	1.7	1.2	1.1	3.1	6.4
Combined (RoBERTa)	1.1	2.9	2.1	3.7	3.4	12.4

Table B.3: Simulated estimator performance on WinoGrande (Infill), WinoGrande (MCQ), and HellaSwag under random and correlated contamination regimes. Values are RMSE against the ground-truth standard model accuracy over 1,000 bootstrap replicates. Estimators using a correctness or memorization predictor reliably adjust for contamination, with the combined estimator using both predictors winning in most settings. Part 1 of 2; see also Table B.4.

Estimator	Random			Correlated (high dose)		
	Low	Mid	High	Easy	Medium	Hard
MMLU						
Naive	1.8	6.4	13.1	0.5	10.3	29.2
IPW (Min-K%++)	1.7	4.5	6.4	7.5	4.1	22.3
Imputation (Llama 3.1)	3.4	2.2	7.5	6.3	5.0	23.1
Imputation (Pythia 6.9B)	3.7	4.5	4.9	18.5	6.6	11.6
Imputation (Qwen3 8B)	3.5	4.3	4.1	15.5	6.1	10.5
Imputation (RoBERTa)	2.7	3.2	3.6	16.8	5.5	13.2
Combined (Llama 3.1)	2.3	3.5	9.6	4.0	6.9	25.4
Combined (Pythia 6.9B)	2.4	1.5	1.6	13.4	2.1	16.4
Combined (Qwen3 8B)	2.3	1.4	1.8	11.2	2.0	15.4
Combined (RoBERTa)	1.9	1.7	2.0	12.5	1.8	17.3
PIQA						
Naive	0.9	3.0	5.3	0.0	0.0	15.9
IPW (Min-K%++)	1.7	1.4	2.5	3.1	3.0	12.9
Imputation (Llama 3.1)	1.8	3.3	5.4	1.4	1.2	15.4
Imputation (Pythia 6.9B)	2.0	1.8	2.3	1.2	2.6	8.5
Imputation (Qwen3 8B)	1.5	1.6	1.6	1.6	4.0	7.7
Imputation (RoBERTa)	4.0	3.4	4.6	1.7	1.8	15.0
Combined (Llama 3.1)	0.9	2.7	4.9	0.6	0.8	15.1
Combined (Pythia 6.9B)	0.8	1.7	2.4	1.0	2.3	9.9
Combined (Qwen3 8B)	0.9	1.5	1.9	1.3	3.2	9.4
Combined (RoBERTa)	1.3	2.3	3.5	2.0	2.0	14.0
PopQA						
Naive	2.4	11.6	24.0	19.2	27.9	25.0
IPW (Min-K%++)	1.3	8.9	18.2	13.2	22.2	19.4
Imputation (Llama 3.1)	1.6	6.3	11.9	7.2	16.0	12.7
Imputation (Pythia 6.9B)	1.7	1.6	1.5	2.6	4.0	2.0
Imputation (Qwen3 8B)	1.5	1.6	1.4	2.3	2.5	1.3
Imputation (RoBERTa)	3.7	2.9	2.9	8.1	2.0	1.7
Combined (Llama 3.1)	1.6	8.2	16.3	11.5	20.4	17.2
Combined (Pythia 6.9B)	1.0	4.8	9.1	6.0	12.8	8.6
Combined (Qwen3 8B)	1.1	5.0	9.1	6.2	11.9	9.2
Combined (RoBERTa)	1.5	3.4	7.7	2.7	11.6	9.0

Table B.4: Simulated estimator performance on MMLU, PIQA, and PopQA under random and correlated contamination regimes. Values are RMSE against the ground-truth standard model accuracy over 1,000 bootstrap replicates. Estimators using a correctness or memorization predictor reliably adjust for contamination, with the combined estimator using both predictors winning in most settings. Part 2 of 2; see also Table B.3.

Benchmark	Random			Correlated (high dose)		
	Low	**Mid**	**High**	**Easy**	**Medium**	**Hard**
WG						
Naive	1.1	3.0	4.6	0.5	0.0	14.0
IPW (Min-K%++)	1.3	2.2	3.1	2.0	1.8	12.7
EPG (Min-K%++)	7.4	4.3	2.1	14.8	3.6	15.3
MMLU						
Naive	1.8	6.4	13.1	0.5	10.3	29.2
IPW (Min-K%++)	1.7	4.5	6.4	7.5	4.1	22.3
EPG-IPW (Min-K%++)	7.4	4.3	2.1	14.8	3.6	15.3
PIQA						
Naive	0.9	3.0	5.3	0.0	0.0	15.9
IPW (Min-K%++)	1.7	1.4	2.5	3.1	3.0	12.9
EPG-IPW (Min-K%++)	9.5	7.7	6.5	10.8	10.7	6.6
HellaSwag						
Naive	0.9	5.1	5.8	0.0	0.5	16.7
IPW (Min-K%++)	1.9	2.5	1.3	5.2	4.9	11.5
EPG-IPW (Min-K%++)	10.0	7.8	8.0	13.3	12.9	5.2

Table B.5: Simulated estimator performance comparing IPW and EPG (spiking-free baseline) using Min-K%++ as the memorization predictor, across all benchmarks. Values are RMSE against the ground-truth standard model accuracy over 1,000 bootstrap replicates, using a test set of size $n = 500$ and contamination rate $\gamma = 0.3$. EPG does not use spiking and chooses the contamination threshold heuristically; its gains over the naive estimator are inconsistent across benchmarks, establishing the necessity of spiking.

Dose	Source	WG	HellaSwag	MMLU	PIQA	PopQA
			Low dose			
	Naive	1.1	0.9	1.8	0.9	2.4
	WG	1.4	2.7	1.7	2.6	2.8
	HellaSwag	1.3	2.8	2.1	2.3	2.9
	MMLU	1.2	1.9	1.5	1.8	1.7
	PIQA	1.3	2.4	1.7	2.3	2.4
	PopQA	1.3	1.9	1.5	1.9	1.6
	Wikipedia	1.7	5.7	3.6	4.4	5.5
			Mid dose			
	Naive	3.0	5.1	6.4	3.0	11.6
	WG	1.9	1.8	5.8	1.0	2.4
	HellaSwag	2.0	1.4	5.7	1.0	1.7
	MMLU	2.3	2.5	5.6	1.1	4.3
	PIQA	2.0	2.0	5.7	0.9	3.0
	PopQA	2.2	2.9	5.8	1.1	5.0
	Wikipedia	1.3	3.5	6.9	3.5	3.9
			High dose			
	Naive	4.6	5.8	13.1	5.3	24.0
	WG	2.0	1.7	5.2	1.5	3.1
	HellaSwag	1.8	2.4	4.0	1.8	1.1
	MMLU	2.4	1.2	4.7	1.2	5.8
	PIQA	2.1	1.5	5.1	1.3	4.0
	PopQA	2.6	1.4	6.7	1.4	8.6
	Wikipedia	1.4	5.4	5.5	3.9	5.8

Table B.6: RMSE of the IPW estimator when the memorization predictor (Min-K%) is transferred across datasets, across low, mid, and high contamination. The memorization predictor is calibrated on the source benchmark and evaluated on all target benchmarks. These simulations are comparable to those of the same settings in Table 4.3. At high contamination, transferred predictors almost always improve over the naive estimator, and predictors calibrated on Wikipedia texts closely match dataset-specific calibration.

Dose	Source	WG	HellaSwag	MMLU	PIQA	PopQA
			Low dose			
	Naive	1.1	0.9	1.8	0.9	2.4
	WG	1.3	1.3	1.8	1.6	1.0
	HellaSwag	1.5	1.9	1.8	2.0	2.1
	MMLU	1.4	1.6	1.7	1.7	1.4
	PIQA	1.4	1.4	1.8	1.7	1.0
	PopQA	1.2	1.1	1.8	1.3	1.3
	Wikipedia	1.1	0.9	1.8	0.9	2.4
			Mid dose			
	Naive	3.0	5.1	6.4	3.0	11.6
	WG	2.2	3.7	5.2	1.6	7.1
	HellaSwag	1.7	2.5	4.2	1.0	3.1
	MMLU	2.0	3.0	4.5	1.2	4.7
	PIQA	2.1	3.5	5.1	1.4	6.5
	PopQA	2.5	4.3	5.7	2.1	8.9
	Wikipedia	3.0	5.1	6.4	3.0	11.6
			High dose			
	Naive	4.6	5.8	13.1	5.3	24.0
	WG	3.1	3.2	9.1	2.8	14.4
	HellaSwag	2.1	1.3	5.3	1.3	5.7
	MMLU	2.4	1.8	6.4	1.7	8.8
	PIQA	3.0	2.9	8.6	2.5	13.2
	PopQA	3.7	4.2	10.7	3.7	18.2
	Wikipedia	4.6	5.8	13.1	5.3	24.0

Table B.7: RMSE of the IPW estimator when the memorization predictor (Min-K%++) is transferred across datasets, across low, mid, and high contamination. The memorization predictor is calibrated on the source benchmark and evaluated on all target benchmarks. These simulations are comparable to those of the same settings in Table 4.3. Min-K%++ is less transferable than Min-K% due to dataset-specific normalization, though at high contamination transferred predictors still generally improve over the naive estimator.

Bibliography

[1] Leo Gao et al. *A framework for few-shot language model evaluation*. Version v0.4.0. Dec. 2023. DOI: 10.5281/zenodo.10256836. URL: https://zenodo.org/records/10256836.

[2] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[3] Martin Ziqiao Ma, Xiang Yue, and Michael Saxon. *The Science of Benchmarking: What's Measured? What's Missed? What's Next?* Tutorial presented at NeurIPS 2025. 2025. URL: https://benchmarking.science/slides.pdf.

[4] Ziang Xiao et al. "Evaluating Evaluation Metrics: A Framework for Analyzing NLG Evaluation Metrics using Measurement Theory". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10967–10982. DOI: 10.18653/v1/2023.emnlp-main.676. URL: https://aclanthology.org/2023.emnlp-main.676/.

[5] Deborah Raji et al. "AI and the Everything in the Whole Wide World Benchmark". In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Ed. by J. Vanschoren and S. Yeung. Vol. 1. 2021. URL: https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/084b6fbb10729ed4da8c3d3f5a3ae7c9-Paper-round2.pdf.

[6] Inbal Magar and Roy Schwartz. "Data Contamination: From Memorization to Exploitation". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 157–165. DOI: 10.18653/v1/2022.acl-short.18. URL: https://aclanthology.org/2022.acl-short.18/.

[7] Oscar Sainz et al. "NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10776–10787. DOI: 10.18653/v1/2023.findings-emnlp.722. URL: https://aclanthology.org/2023.findings-emnlp.722/.

[8] Yucheng Li et al. "An Open-Source Data Contamination Report for Large Language Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA:

Association for Computational Linguistics, Nov. 2024, pp. 528–541. DOI: 10.18653/v1/2024.findings-emnlp.30. URL: https://aclanthology.org/2024.findings-emnlp.30/.

[9] Yanai Elazar et al. "What's In My Big Data?" In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=RvfPnOkPV4.

[10] Nicholas Carlini et al. "Extracting Training Data from Large Language Models". In: *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 2633–2650. ISBN: 978-1-939133-24-3. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

[11] Jamie Hayes et al. "Measuring memorization in language models via probabilistic extraction". In: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Ed. by Luis Chiruzzo, Alan Ritter, and Lu Wang. Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 9266–9291. ISBN: 979-8-89176-189-6. DOI: 10.18653/v1/2025.naacl-long.469. URL: https://aclanthology.org/2025.naacl-long.469/.

[12] Nikhil Kandpal, Eric Wallace, and Colin Raffel. "Deduplicating Training Data Mitigates Privacy Risks in Language Models". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 10697–10707. URL: https://proceedings.mlr.press/v162/kandpal22a.html.

[13] Kushal Tirumala et al. "Memorization without overfitting: Analyzing the training dynamics of large language models". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 38274–38290.

[14] Fabio Petroni et al. "Language Models as Knowledge Bases?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2463–2473. DOI: 10.18653/v1/D19-1250. URL: https://aclanthology.org/D19-1250/.

[15] Vitaly Feldman and Chiyuan Zhang. "What neural networks memorize and why: discovering the long tail via influence estimation". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[16] Valentin Hartmann et al. *SoK: Memorization in General-Purpose Large Language Models*. 2023. arXiv: 2310.18362 [cs.CL]. URL: https://arxiv.org/abs/2310.18362.

[17] Peter Henderson et al. "Foundation Models and Fair Use". In: *Journal of Machine Learning Research* 24.400 (2023), pp. 1–79. URL: http://jmlr.org/papers/v24/23-0569.html.

[18] Hannah Brown et al. "What Does it Mean for a Language Model to Preserve Privacy?" In: *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*. ACM, 2022, pp. 2280–2292. DOI: 10.1145/3531146.3534642. URL: https://doi.org/10.1145/3531146.3534642.

[19] Johnny Wei, Ryan Wang, and Robin Jia. "Proving membership in LLM pretraining data via data watermarks". In: *Findings of the Association for Computational Linguistics: ACL 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 13306–13320. DOI: 10.18653/v1/2024.findings-acl.788. URL: https://aclanthology.org/2024.findings-acl.788/.

[20] Johnny Tian-Zheng Wei et al. "Interrogating LLM design under copyright law". In: *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '25. New York, NY, USA: Association for Computing Machinery, 2025, 3030–3045. ISBN: 9798400714825. DOI: 10.1145/3715275.3732193. URL: https://doi.org/10.1145/3715275.3732193.

[21] Johnny Tian-Zheng Wei, Tom Kocmi, and Christian Federmann. "Searching for a Higher Power in the Human Evaluation of MT". In: *Proceedings of the Seventh Conference on Machine Translation (WMT)*. Ed. by Philipp Koehn et al. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 129–139. URL: https://aclanthology.org/2022.wmt-1.7/.

[22] Peng Ding. *A First Course in Causal Inference*. 2023. arXiv: 2305.18793 [stat.ME]. URL: https://arxiv.org/abs/2305.18793.

[23] Joshua D. Angrist and Jörn-Steffen Pischke. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, 2009. ISBN: 9780691120348. URL: http://www.jstor.org/stable/j.ctvcm4j72 (visited on 09/26/2024).

[24] Yonatan Oren et al. "Proving Test Set Contamination in Black-Box Language Models". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=KS8mIvetg2.

[25] Sally Zhu et al. *Independence Tests for Language Models*. 2025. URL: https://openreview.net/forum?id=leJWwts8P9.

[26] Thomas Steinke, Milad Nasr, and Matthew Jagielski. "Privacy auditing with one (1) training run". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS '23. New Orleans, LA, USA: Curran Associates Inc., 2023.

[27] Brad A. Greenberg. "Rethinking Technology Neutrality". In: *Minnesota Law Review* 100 (2016). Available at SSRN: https://ssrn.com/abstract=2748932, p. 1495.

[28] U.S. Constitution. *17 U.S.C. § 106: Exclusive Rights in Copyrighted Works*. United States Code. This section grants copyright owners exclusive rights to reproduce, prepare derivative works, distribute copies, perform publicly, display publicly, and, in the case of sound recordings, perform publicly by means of digital audio transmission. 2024. URL: https://www.law.cornell.edu/uscode/text/17/106.

[29] *American Broadcasting Companies, Inc. v. Aereo, Inc.* Available at: https://www.supremecourt.gov/opinions/13pdf/13-461_l537.pdf. 2014.

[30] Matthew Sag. "Copyright Safety for Generative AI". In: *Houston Law Review* 61.2 (2023). DOI: 10.2139/ssrn.4438593. URL: https://ssrn.com/abstract=4438593.

[31] Andrew D. Selbst, Suresh Venkatasubramanian, and I. Elizabeth Kumar. "Deconstructing Design Decisions: Why Courts Must Interrogate Machine Learning and Other Technologies". In: *Ohio State Law Journal* 85 (2024). UCLA School of Law, Public Law Research Paper No. 23-22, p. 415. URL: https://ssrn.com/abstract=4564304.

[32] A. Feder Cooper and James Grimmelmann. "The Files are in the Computer: On Copyright, Memorization, and Generative AI". In: *Chicago-Kent Law Review* 100 (2025). Cornell Legal Studies Research Paper No. 24-30, pp. 141–219. URL: https://ssrn.com/abstract=4803118.

[33] U.S. Constitution. *Article I, Section 8, Clause 8*. "To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries". 1787.

[34] Stella Biderman et al. "Pythia: A suite for analyzing large language models across training and scaling". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 2397–2430.

[35] *The New York Times Co. v. Microsoft Corp.* No. 1:2023cv11195, 17 U.S.C. § 501 Copyright Infringement (S.D.N.Y. filed Dec. 27, 2023). 2023.

[36] Ashish Vaswani et al. "Attention is all you need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, 6000–6010. ISBN: 9781510860964.

[37] Leo Gao et al. *The Pile: An 800GB Dataset of Diverse Text for Language Modeling*. 2020. arXiv: 2101.00027 [cs.CL]. URL: https://arxiv.org/abs/2101.00027.

[38] Suchin Gururangan et al. "Whose Language Counts as High Quality? Measuring Language Ideologies in Text Data Selection". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 2562–2580. DOI: 10.18653/v1/2022.emnlp-main.165. URL: https://aclanthology.org/2022.emnlp-main.165.

[39] Abeba Birhane et al. "The Values Encoded in Machine Learning Research". In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '22. Seoul, Republic of Korea: Association for Computing Machinery, 2022, 173–184. ISBN: 9781450393522. DOI: 10.1145/3531146.3533083. URL: https://doi.org/10.1145/3531146.3533083.

[40] Weijia Shi et al. "Detecting Pretraining Data from Large Language Models". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=zWqr3MQuNs.

[41] Jing Huang, Diyi Yang, and Christopher Potts. *Demystifying Verbatim Memorization in Large Language Models*. 2024. arXiv: 2407.17817 [cs.CL]. URL: https://arxiv.org/abs/2407.17817.

[42] Aaron Grattafiori et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[43] Edward Lee. *Master List of Lawsuits v. AI: ChatGPT, OpenAI, Microsoft, Meta, MidJourney, Other AI Cos.* Accessed: 2025-9-9. 2024. URL: https://chatgptiseatingtheworld.com/.

[44] OpenAI. *OpenAI and Journalism*. Accessed: 2025-01-21. 2025. URL: https://openai.com/index/openai-and-journalism/.

[45] Daphne Ippolito et al. "Preventing Generation of Verbatim Memorization in Language Models Gives a False Sense of Privacy". In: *Proceedings of the 16th International Natural Language Generation Conference*. Ed. by C. Maria Keet, Hung-Yi Lee, and Sina Zarrieß. Prague, Czechia: Association for Computational Linguistics, Sept. 2023, pp. 28–53. DOI: 10.18653/v1/2023.inlg-main.3. URL: https://aclanthology.org/2023.inlg-main.3/.

[46] Gemini Team et al. "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context". In: *arXiv preprint arXiv:2403.05530* (2024).

[47] Kyle O'Brien et al. "Deep Ignorance: Filtering Pretraining Data Builds Tamper-Resistant Safeguards into Open-Weight LLMs". In: *The Fourteenth International Conference on Learning Representations*. 2026. URL: https://openreview.net/forum?id=xcf0QcTcGS.

[48] Amer Sinha et al. *VaultGemma: A Differentially Private Gemma Model*. 2025. arXiv: 2510.15001 [cs.CR]. URL: https://arxiv.org/abs/2510.15001.

[49] A. Feder Cooper et al. *Machine Unlearning Doesn't Do What You Think: Lessons for Generative AI Policy, Research, and Practice*. 2024. arXiv: 2412.06966 [cs.LG]. URL: https://arxiv.org/abs/2412.06966.

[50] Chiyuan Zhang et al. "Counterfactual memorization in neural language models". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS '23. New Orleans, LA, USA: Curran Associates Inc., 2023.

[51] Zeyuan Allen-Zhu and Yuanzhi Li. "Physics of language models: part 3.1, knowledge storage and extraction". In: *Proceedings of the 41st International Conference on Machine Learning*. ICML'24. Vienna, Austria: JMLR.org, 2024.

[52] John X. Morris et al. *How much do language models memorize?* 2025. arXiv: 2505.24832 [cs.CL]. URL: https://arxiv.org/abs/2505.24832.

[53] USVSN Sai Prashanth et al. "Recite, Reconstruct, Recollect: Memorization in LMs as a Multifaceted Phenomenon". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=3E8YNv1HjU.

[54] Pietro Lesci et al. "Causal Estimation of Memorisation Profiles". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15616–15635. DOI: 10.18653/v1/2024.acl-long.834. URL: https://aclanthology.org/2024.acl-long.834/.

[55] Johnny Wei et al. "Hubble: a Model Suite to Advance the Study of LLM Memorization". In: *The Fourteenth International Conference on Learning Representations*. 2026. URL: https://openreview.net/forum?id=ZfdnZhOP0k.

[56] Stella Biderman et al. "Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling". In: *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 2397–2430. URL: https://proceedings.mlr.press/v202/biderman23a.html.

[57] Michael Duan et al. "Do Membership Inference Attacks Work on Large Language Models?" In: *First Conference on Language Modeling*. 2024. URL: https://openreview.net/forum?id=av0D19pSkU.

[58] Aravind Krishnan, Siva Reddy, and Marius Mosbach. *Not All Data Are Unlearned Equally*. 2025. arXiv: 2504.05058 [cs.CL]. URL: https://arxiv.org/abs/2504.05058.

[59] Ali Satvaty, Suzan Verberne, and Fatih Turkmen. *Undesirable Memorization in Large Language Models: A Survey*. 2025. arXiv: 2410.02650 [cs.CL]. URL: https://arxiv.org/abs/2410.02650.

[60] Shayne Longpre et al. "A large-scale audit of dataset licensing and attribution in AI". In: *Nature Machine Intelligence* 6.8 (Aug. 2024), pp. 975–987. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00878-8. URL: https://doi.org/10.1038/s42256-024-00878-8.

[61] Rachel Hong et al. *A Common Pool of Privacy Problems: Legal and Technical Lessons from a Large-Scale Web-Scraped Machine Learning Dataset*. 2025. arXiv: 2506.17185 [cs.CR]. URL: https://arxiv.org/abs/2506.17185.

[62] Alon Jacovi et al. "Stop Uploading Test Data in Plain Text: Practical Strategies for Mitigating Data Contamination by Evaluation Benchmarks". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5075–5084. DOI: 10.18653/v1/2023.emnlp-main.308. URL: https://aclanthology.org/2023.emnlp-main.308/.

[63] Katherine Lee, A. Feder Cooper, and James Grimmelmann. *Talkin' 'Bout AI Generation: Copyright and the Generative-AI Supply Chain*. 2024. arXiv: 2309.08133 [cs.CY]. URL: https://arxiv.org/abs/2309.08133.

[64] U.S. Copyright Office. *Copyright and Artificial Intelligence, Part 3: Generative AI Training*. Tech. rep. Pre-Publication Version. U.S. Copyright Office, 2025. URL: https://www.copyright.gov/ai/.

[65] U.S. Constitution. *17 U.S.C. § 107: Limitations on Exclusive Rights—Fair Use*. United States Code. This section outlines the fair use doctrine, permitting certain uses of copyrighted works for purposes such as criticism, comment, news reporting, teaching, scholarship, or research. It specifies four factors to consider in determining fair use: (1) the purpose and character of the use; (2) the nature of the copyrighted work; (3) the amount and substantiality of the portion used; and (4) the effect of the use upon the potential market for or value of the copyrighted work. 2024. URL: https://www.law.cornell.edu/uscode/text/17/107.

[66] Boyi Wei et al. "Evaluating Copyright Takedown Methods for Language Models". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 139114–139150. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/faed4276b52ef762879db4142655c699-Paper-Datasets_and_Benchmarks_Track.pdf.

[67] Kent Chang et al. "Speak, Memory: An Archaeology of Books Known to ChatGPT/GPT-4". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 7312–7327. DOI: 10.18653/v1/2023.emnlp-main.453. URL: https://aclanthology.org/2023.emnlp-main.453.

[68] A. Feder Cooper et al. *Extracting Memorized Pieces of (Copyrighted) Books from Open-Weight Language Models*. SSRN Working Paper No. 5262084, Stanford Public Law Working Paper; WVU College of Law Research Paper No. 2025-005. Posted 21 May 2025; Last revised 11 July 2025. Apr. 2025. DOI: 10.2139/ssrn.5262084. URL: https://ssrn.com/abstract=5262084.

[69] Avi Schwarzschild et al. "Rethinking LLM Memorization through the Lens of Adversarial Compression". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 56244–56267. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/66453d578afae006252d2ea090e151c9-Paper-Conference.pdf.

[70] Martin Gerlach and Francesc Font-Clos. *A standardized Project Gutenberg corpus for statistical analysis of natural language and quantitative linguistics*. 2018. arXiv: 1812.08092 [cs.CL]. URL: https://arxiv.org/abs/1812.08092.

[71] Xinyi Wang et al. "Generalization v.s. Memorization: Tracing Language Models' Capabilities Back to Pretraining Data". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=IQxBDLmVpT.

[72] John Kirchenbauer et al. "LMD3: Language Model Data Density Dependence". In: *First Conference on Language Modeling*. 2024. URL: https://openreview.net/forum?id=eGCw1UVOhk.

[73] Bill Dolan and Chris Brockett. "Automatically Constructing a Corpus of Sentential Paraphrases". In: *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing, 2005. URL: https://www.microsoft.com/en-us/research/publication/automatically-constructing-a-corpus-of-sentential-paraphrases/.

[74] Yuan Zhang, Jason Baldridge, and Luheng He. "PAWS: Paraphrase Adversaries from Word Scrambling". In: *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 1298–1308. DOI: 10.18653/v1/N19-1131. URL: https://aclanthology.org/N19-1131/.

[75] Tong Chen et al. "CopyBench: Measuring Literal and Non-Literal Reproduction of Copyright-Protected Text in Language Model Generation". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 15134–15158. DOI: 10.18653/v1/2024.emnlp-main.844. URL: https://aclanthology.org/2024.emnlp-main.844/.

[76] Jaechul Roh et al. *Bob's Confetti: Phonetic Memorization Attacks in Music and Video Generation*. 2025. arXiv: 2507.17937 [cs.SD]. URL: https://arxiv.org/abs/2507.17937.

[77] European Union. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. Official Journal of the European Union, L 119, 4 May 2016, p. 1–88. Accessed: 2025-09-08. 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[78] State of California. *California Consumer Privacy Act of 2018*. https://oag.ca.gov/privacy/ccpa. Cal. Civ. Code §§ 1798.100–1798.199. 2018.

[79] Joseph P. Near et al. *Guidelines for Evaluating Differential Privacy Guarantees*. Tech. rep. NIST Special Publication 800-226. National Institute of Standards and Technology, 2023. DOI: 10.6028/NIST.SP.800-226. URL: https://doi.org/10.6028/NIST.SP.800-226.

[80] Lucas Bourtoule et al. "Machine unlearning". English (US). In: *Proceedings - 2021 IEEE Symposium on Security and Privacy, SP 2021*. Proceedings - IEEE Symposium on Security and Privacy. United States: Institute of Electrical and Electronics Engineers Inc., May 2021, pp. 141–159. DOI: 10.1109/SP40001.2021.00019.

[81] Kevin Meng et al. "Locating and Editing Factual Associations in GPT". In: *Advances in Neural Information Processing Systems* 36 (2022). arXiv:2202.05262.

[82] Nils Lukas et al. "Analyzing leakage of personally identifiable information in language models". In: *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2023, pp. 346–363.

[83] Ashwinee Panda et al. "Teach llms to phish: Stealing private information from language models". In: *arXiv preprint arXiv:2403.00871* (2024).

[84] Jaydeep Borkar et al. "Privacy Ripple Effects from Adding or Removing Personal Information in Language Model Training". In: *arXiv preprint arXiv:2502.15680* (2025).

[85] Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. "Are Large Pre-Trained Language Models Leaking Your Personal Information?" In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. 2022, pp. 2038–2047.

[86] Shenglai Zeng et al. "Exploring Memorization in Fine-tuned Language Models". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2024, pp. 3917–3948.

[87] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. "YAGO 4: A Reason-able Knowledge Base". In: *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings*. Heraklion, Crete, Greece: Springer-Verlag, 2020, 583–596. ISBN: 978-3-030-49460-5. DOI: 10.1007/978-3-030-49461-2_34. URL: https://doi.org/10.1007/978-3-030-49461-2_34.

[88] Ildikó Pilán et al. "The text anonymization benchmark (tab): A dedicated corpus and evaluation framework for text anonymization". In: *Computational Linguistics* 48.4 (2022), pp. 1053–1101.

[89] Hanna Yukhymenko et al. "A synthetic dataset for personal attribute inference". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems*. NIPS '24. Vancouver, BC, Canada: Curran Associates Inc., 2025. ISBN: 9798331314385.

[90] Saizheng Zhang et al. "Personalizing Dialogue Agents: I have a dog, do you have pets too?" In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 2204–2213. DOI: 10.18653/v1/P18-1205. URL: https://aclanthology.org/P18-1205/.

[91] Federal Trade Commission. *FTC Announces Crackdown on Deceptive AI Claims and Schemes*. https://www.ftc.gov/news-events/news/press-releases/2024/09/ftc-announces-crackdown-deceptive-ai-claims-schemes. Press Release. Sept. 2024.

[92] Kawin Ethayarajh and Dan Jurafsky. "Utility is in the Eye of the User: A Critique of NLP Leaderboards". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 4846–4853. DOI: 10.1865

3/v1/2020.emnlp-main.393. URL: https://aclanthology.org/2020.emnlp-main.393/.

[93] Shahriar Golchin and Mihai Surdeanu. "Time Travel in LLMs: Tracing Data Contamination in Large Language Models". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=2Rwq6c3tvr.

[94] Yujuan Fu et al. "Does Data Contamination Detection Work (Well) for LLMs? A Survey and Evaluation on Detection Assumptions". In: *Findings of the Association for Computational Linguistics: NAACL 2025*. Ed. by Luis Chiruzzo, Alan Ritter, and Lu Wang. Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 5235–5256. ISBN: 979-8-89176-195-7. DOI: 10.18653/v1/2025.findings-naacl.291. URL: https://aclanthology.org/2025.findings-naacl.291/.

[95] Aaditya K. Singh et al. *Evaluation data contamination in LLMs: how do we measure it and (when) does it matter?* 2024. arXiv: 2411.03923 [cs.CL]. URL: https://arxiv.org/abs/2411.03923.

[96] Jesse Dodge et al. "Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1286–1305. DOI: 10.18653/v1/2021.emnlp-main.98. URL: https://aclanthology.org/2021.emnlp-main.98/.

[97] Minhao Jiang et al. *Investigating Data Contamination for Pre-training Language Models*. 2024. arXiv: 2401.06059 [cs.CL]. URL: https://arxiv.org/abs/2401.06059.

[98] Akshara Prabhakar, Thomas L. Griffiths, and R. Thomas McCoy. "Deciphering the Factors Influencing the Efficacy of Chain-of-Thought: Probability, Memorization, and Noisy Reasoning". In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 3710–3724. DOI: 10.18653/v1/2024.findings-emnlp.212. URL: https://aclanthology.org/2024.findings-emnlp.212/.

[99] Changmao Li and Jeffrey Flanigan. "Task Contamination: Language Models May Not Be Few-Shot Anymore". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.16 (2024), pp. 18471–18480. DOI: 10.1609/aaai.v38i16.29808. URL: https://ojs.aaai.org/index.php/AAAI/article/view/29808.

[100] Davide Testa, Emmanuele Chersoni, and Alessandro Lenci. "We Understand Elliptical Sentences, and Language Models should Too: A New Dataset for Studying Ellipsis

and its Interaction with Thematic Fit". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 3340–3353. DOI: 10.18653/v1/2023.acl-long.188. URL: https://aclanthology.org/2023.acl-long.188/.

[101] Xiaoyu Tong et al. "Metaphor Understanding Challenge Dataset for LLMs". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 3517–3536. DOI: 10.18653/v1/2024.acl-long.193. URL: https://aclanthology.org/2024.acl-long.193/.

[102] Dirk Groeneveld et al. "OLMo: Accelerating the Science of Language Models". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15789–15809. DOI: 10.18653/v1/2024.acl-long.841. URL: https://aclanthology.org/2024.acl-long.841/.

[103] Project Apertus et al. *Apertus: Democratizing Open and Compliant LLMs for Global Language Environments*. 2025. arXiv: 2509.14233 [cs.CL]. URL: https://arxiv.org/abs/2509.14233.

[104] Zhengzhong Liu et al. *LLM360: Towards Fully Transparent Open-Source LLMs*. 2023. arXiv: 2312.06550 [cs.CL]. URL: https://arxiv.org/abs/2312.06550.

[105] Apoorv Khandelwal et al. "$100K or 100 Days: Trade-offs when Pre-Training with Academic Resources". In: *Second Conference on Language Modeling*. 2025. URL: https://openreview.net/forum?id=EFxC34XbDh.

[106] Jordan Hoffmann et al. "Training compute-optimal large language models". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS '22. New Orleans, LA, USA: Curran Associates Inc., 2022. ISBN: 9781713871088.

[107] Jeffrey Li et al. "DataComp-LM: In search of the next generation of training sets for language models". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 14200–14282. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/19e4ea30ded58259665db375885e412-Paper-Datasets_and_Benchmarks_Track.pdf.

[108] Mohammad Aflah Khan et al. *TokenSmith: Streamlining Data Editing, Search, and Inspection for Large-Scale Language Model Training and Interpretability*. 2025. arXiv: 2507.19419 [cs.CL]. URL: https://arxiv.org/abs/2507.19419.

[109] Danny Hernandez et al. *Scaling Laws and Interpretability of Learning from Repeated Data*. 2022. arXiv: 2205.10487 [cs.LG]. URL: https://arxiv.org/abs/2205.10487.

[110] Nostalgebraist. *Interpreting GPT: the Logit Lens*. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. LessWrong blog post. 2020.

[111] Nora Belrose et al. *Eliciting Latent Predictions from Transformers with the Tuned Lens*. 2025. arXiv: 2303.08112 [cs.LG]. URL: https://arxiv.org/abs/2303.08112.

[112] Reza Shokri et al. "Membership Inference Attacks Against Machine Learning Models". In: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, pp. 3–18. DOI: 10.1109/SP.2017.41.

[113] Pranav Rajpurkar, Robin Jia, and Percy Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 784–789. DOI: 10.18653/v1/P18-2124. URL: https://aclanthology.org/P18-2124/.

[114] Sebastian Bordt et al. "How Much Can We Forget about Data Contamination?" In: *Forty-second International Conference on Machine Learning*. 2025. URL: https://openreview.net/forum?id=Pf0PaYS9KG.

[115] Katherine Lee et al. "Deduplicating Training Data Makes Language Models Better". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8424–8445. DOI: 10.18653/v1/2022.acl-long.577. URL: https://aclanthology.org/2022.acl-long.577/.

[116] Matthew Jagielski et al. "Measuring Forgetting of Memorized Training Examples". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=7bJizxLKrR.

[117] Hoyeon Chang et al. "How Do Large Language Models Acquire Factual Knowledge During Pretraining?" In: *Advances in Neural Information Processing Systems*. 2024.

[118] Yash More, Prakhar Ganesh, and Golnoosh Farnadi. *Towards More Realistic Extraction Attacks: An Adversarial Perspective*. 2025. arXiv: 2407.02596 [cs.CR]. URL: https://arxiv.org/abs/2407.02596.

[119] Andrew Ilyas et al. “Datamodels: Understanding Predictions with Data and Data with Predictions”. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 9525–9587. URL: https://proceedings.mlr.press/v162/ilyas22a.html.

[120] Sijia Liu et al. *Rethinking Machine Unlearning for Large Language Models*. 2024. arXiv: 2402.08787 [cs.LG]. URL: https://arxiv.org/abs/2402.08787.

[121] Matthieu Meeus et al. “ SoK: Membership Inference Attacks on LLMs are Rushing Nowhere (and How to Fix It) ”. In: *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. Los Alamitos, CA, USA: IEEE Computer Society, Apr. 2025, pp. 385–401. DOI: 10.1109/SaTML64287.2025.00028. URL: https://doi.ieeecomputersociety.org/10.1109/SaTML64287.2025.00028.

[122] Ali Naseh and Niloofar Mireshghallah. *Synthetic Data Can Mislead Evaluations: Membership Inference as Machine Text Detection*. 2025. arXiv: 2501.11786 [cs.CL]. URL: https://arxiv.org/abs/2501.11786.

[123] Vineeth Dorna et al. *OpenUnlearning: Accelerating LLM Unlearning via Unified Benchmarking of Methods and Metrics*. 2025. arXiv: 2506.12618 [cs.CL]. URL: https://arxiv.org/abs/2506.12618.

[124] Samuel Yeom et al. “ Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting ”. In: *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. Los Alamitos, CA, USA: IEEE Computer Society, July 2018, pp. 268–282. DOI: 10.1109/CSF.2018.00027. URL: https://doi.ieeecomputersociety.org/10.1109/CSF.2018.00027.

[125] Jingyang Zhang et al. “Min-K%++: Improved Baseline for Pre-Training Data Detection from Large Language Models”. In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=ZGkfoufDaU.

[126] Pratyush Maini et al. “TOFU: A Task of Fictitious Unlearning for LLMs”. In: *First Conference on Language Modeling*. 2024. URL: https://openreview.net/forum?id=B41hNBoWLo.

[127] Weijia Shi et al. “MUSE: Machine Unlearning Six-Way Evaluation for Language Models”. In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=TArmA033BU.

[128] Nathaniel Li et al. “The WMDP Benchmark: Measuring and Reducing Malicious Use with Unlearning”. In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov et al. Vol. 235. Proceedings of Machine Learn-

ing Research. PMLR, 2024, pp. 28525–28550. URL: https://proceedings.mlr.press/v235/li24bc.html.

[129] Andy Zou et al. "Improving Alignment and Robustness with Circuit Breakers". In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024. URL: https://openreview.net/forum?id=IbIB8SBKFV.

[130] Puning Yang et al. "Exploring Criteria of Loss Reweighting to Enhance LLM Unlearning". In: *Forty-second International Conference on Machine Learning*. 2025. URL: https://openreview.net/forum?id=mGOugCZlAq.

[131] Stephen Merity et al. "Pointer Sentinel Mixture Models". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=Byj72udxe.

[132] Rohit Gandikota et al. *Erasing Conceptual Knowledge from Language Models*. 2025. arXiv: 2410.02760 [cs.CL]. URL: https://arxiv.org/abs/2410.02760.

[133] Mor Geva et al. "Transformer Feed-Forward Layers Are Key-Value Memories". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 5484–5495. DOI: 10.18653/v1/2021.emnlp-main.446. URL: https://aclanthology.org/2021.emnlp-main.446/.

[134] Damai Dai et al. "Knowledge Neurons in Pretrained Transformers". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8493–8502. DOI: 10.18653/v1/2022.acl-long.581. URL: https://aclanthology.org/2022.acl-long.581/.

[135] Weijia Shi et al. *FlexOlmo: Open Language Models for Flexible Data Use*. 2025. arXiv: 2507.07024 [cs.CL]. URL: https://arxiv.org/abs/2507.07024.

[136] Gaurav R. Ghosal, Pratyush Maini, and Aditi Raghunathan. *Memorization Sinks: Isolating Memorization during LLM Training*. 2025. arXiv: 2507.09937 [cs.LG]. URL: https://arxiv.org/abs/2507.09937.

[137] Pratyush Maini et al. "Can neural network memorization be localized?" In: *Proceedings of the 40th International Conference on Machine Learning*. ICML'23. Honolulu, Hawaii, USA: JMLR.org, 2023.

[138] Ting-Yun Chang, Jesse Thomason, and Robin Jia. "Do Localization Methods Actually Localize Memorized Data in LLMs? A Tale of Two Benchmarks". In: *Proceedings*

of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 3190–3211. DOI: 10.18653/v1/2024.naacl-long.176. URL: https://aclanthology.org/2024.naacl-long.176/.

[139] Stella Biderman et al. "Emergent and Predictable Memorization in Large Language Models". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=Iq0DvhB4Kf.

[140] Ashwinee Panda et al. "Privacy Auditing of Large Language Models". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=60Vd7QOXlM.

[141] Roger Grosse et al. *Studying Large Language Model Generalization with Influence Functions*. 2023. arXiv: 2308.03296 [cs.LG]. URL: https://arxiv.org/abs/2308.03296.

[142] Ting-Yun Chang et al. *Why Do Some Inputs Break Low-Bit LLM Quantization?* 2025. arXiv: 2506.12044 [cs.LG]. URL: https://arxiv.org/abs/2506.12044.

[143] Tanishq Kumar et al. "Scaling Laws for Precision". eng. In: *International Conference on Learning Representations (ICLR) (Oral)* (2025).

[144] Alexandra Souly et al. *Poisoning Attacks on LLMs Require a Near-constant Number of Poison Samples*. 2025. arXiv: 2510.07192 [cs.LG]. URL: https://arxiv.org/abs/2510.07192.

[145] Niloofar Mireshghallah and Tianshi Li. *Position: Privacy Is Not Just Memorization!* 2025. arXiv: 2510.01645 [cs.CR]. URL: https://arxiv.org/abs/2510.01645.

[146] Daniela Gottesman et al. *LMEnt: A Suite for Analyzing Knowledge in Language Models from Pretraining Data to Representations*. 2025. arXiv: 2509.03405 [cs.CL]. URL: https://arxiv.org/abs/2509.03405.

[147] Donald B. Rubin. "Causal Inference Using Potential Outcomes: Design, Modeling, Decisions". In: *Journal of the American Statistical Association* 100.469 (2005), pp. 322–331. ISSN: 01621459. URL: http://www.jstor.org/stable/27590541 (visited on 03/05/2026).

[148] Reza Shokri et al. "Membership Inference Attacks Against Machine Learning Models". In: *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 2017, pp. 3–18. DOI: 10.1109/SP.2017.41. URL: https://doi.org/10.1109/SP.2017.41.

[149] Jasper Dekoninck, Mark Niklas Müller, and Martin Vechev. "ConStat: performance-based contamination detection in large language models". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems*. NIPS '24. Vancouver, BC, Canada: Curran Associates Inc., 2024. ISBN: 9798331314385.

[150] Rylan Schaeffer et al. *Quantifying the Effect of Test Set Contamination on Generative Evaluations*. 2026. arXiv: 2601.04301 [cs.LG]. URL: https://arxiv.org/abs/2601.04301.

[151] Shahriar Golchin and Mihai Surdeanu. "Data Contamination Quiz: A Tool to Detect and Estimate Contamination in Large Language Models". In: *Transactions of the Association for Computational Linguistics* 13 (July 2025), pp. 809–830. ISSN: 2307-387X. DOI: 10.1162/TACL.a.20. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/TACL.a.20/2540087/tacl.a.20.pdf. URL: https://doi.org/10.1162/TACL.a.20.

[152] John C. Platt. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". In: *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[153] Chuan Guo et al. "On calibration of modern neural networks". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, 1321–1330.

[154] Behzad Mehrbakhsh et al. "Confounders in Instance Variation for the Analysis of Data Contamination". In: *Proceedings of the 1st Workshop on Data Contamination (CONDA)*. Ed. by Oscar Sainz et al. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 13–21. DOI: 10.18653/v1/2024.conda-1.2. URL: https://aclanthology.org/2024.conda-1.2/.

[155] Michael Saxon et al. "Benchmarks as Microscopes: A Call for Model Metrology". In: *First Conference on Language Modeling*. 2024. URL: https://openreview.net/forum?id=bttKwCZDkm.

[156] Sebastian Bordt and Martin Pawelczyk. "Train Once, Answer All: Many Pretraining Experiments for the Cost of One". In: *The Fourteenth International Conference on Learning Representations*. 2026. URL: https://openreview.net/forum?id=EoBmdFujak.

[157] Art B. Owen. *Monte Carlo theory, methods and examples*. https://artowen.su.domains/mc/, 2013.

[158] Hlynur Davíð Hlynsson. *A Tutorial on Doubly Robust Learning for Causal Inference*. 2024. arXiv: 2406.00853 [stat.ML]. URL: https://arxiv.org/abs/2406.00853.

[159] Keisuke Sakaguchi et al. "WinoGrande: an adversarial winograd schema challenge at scale". In: *Commun. ACM* 64.9 (Aug. 2021), 99–106. ISSN: 0001-0782. DOI: 10.1145/3474381. URL: https://doi.org/10.1145/3474381.

[160] Dan Hendrycks et al. "Measuring Massive Multitask Language Understanding". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=d7KBjmI3GmQ.

[161] Rowan Zellers et al. "HellaSwag: Can a Machine Really Finish Your Sentence?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4791–4800. DOI: 10.18653/v1/P19-1472. URL: https://aclanthology.org/P19-1472/.

[162] Yonatan Bisk et al. "PIQA: Reasoning about Physical Commonsense in Natural Language". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05 (2020), pp. 7432–7439. DOI: 10.1609/aaai.v34i05.6239. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6239.

[163] Alex Mallen et al. "When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 9802–9822. DOI: 10.18653/v1/2023.acl-long.546. URL: https://aclanthology.org/2023.acl-long.546/.

[164] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. 1st. Chapman and Hall/CRC, 1994. DOI: 10.1201/9780429246593.

[165] Samuel Yeom et al. "Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning". In: *J. Comput. Secur.* 28.1 (Jan. 2020), 35–70. ISSN: 0926-227X. DOI: 10.3233/JCS-191362. URL: https://doi.org/10.3233/JCS-191362.

[166] Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. "Understanding Dataset Difficulty with $\mathcal{V}$-Usable Information". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 5988–6008. URL: https://proceedings.mlr.press/v162/ethayarajh22a.html.

[167] Sang T. Truong et al. "Reliable and Efficient Amortized Model-based Evaluation". In: *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. Ed. by Aarti Singh et al. Vol. 267. Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL: https://proceedings.mlr.press/v267/truong25c.html.

[168] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL]. URL: https://arxiv.org/abs/1907.11692.

[169] Qwen Team. *Qwen3 Technical Report*. 2025. arXiv: 2505.09388 [cs.CL]. URL: https://arxiv.org/abs/2505.09388.

[170] Benjamin Recht et al. "Do ImageNet Classifiers Generalize to ImageNet?" In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 5389–5400. URL: https://proceedings.mlr.press/v97/recht19a.html.

[171] Anka Reuel et al. "Open Problems in Technical AI Governance". In: *Transactions on Machine Learning Research* (2025). Survey Certification. ISSN: 2835-8856. URL: https://openreview.net/forum?id=1nO4qFMiS0.

[172] Stephen Casper et al. "Black-Box Access is Insufficient for Rigorous AI Audits". In: *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '24. Rio de Janeiro, Brazil: Association for Computing Machinery, 2024, 2254–2272. ISBN: 9798400704505. DOI: 10.1145/3630106.3659037. URL: https://doi.org/10.1145/3630106.3659037.

[173] Drew Keller et al. *Practices for Automated Benchmark Evaluations of Language Models*. Initial Public Draft NIST AI 800-2. National Institute of Standards and Technology, Jan. 2026. DOI: 10.6028/NIST.AI.800-2.ipd. URL: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.800-2.ipd.pdf.

[174] Blake Bullwinkel et al. *The Trigger in the Haystack: Extracting and Reconstructing LLM Backdoor Triggers*. 2026. arXiv: 2602.03085 [cs.CR]. URL: https://arxiv.org/abs/2602.03085.

[175] Jun Yan et al. "Rethinking Backdoor Detection Evaluation for Language Models". In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Ed. by Christos Christodoulopoulos et al. Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 6228–6239. ISBN: 979-8-89176-332-6. DOI: 10.18653/v1/2025.emnlp-main.318. URL: https://aclanthology.org/2025.emnlp-main.318/.

[176] John Hewitt and Percy Liang. "Designing and Interpreting Probes with Control Tasks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2733–2743. DOI: 10.18653/v1/D19-1275. URL: https://aclanthology.org/D19-1275/.

[177] Kentaro Toyama. *Geek heresy: Rescuing social change from the cult of technology*. PublicAffairs, 2015.

[178] Sarah H. Cen and Rohan Alur. *From Transparency to Accountability and Back: A Discussion of Access and Evidence in AI Auditing*. 2024. arXiv: 2410.04772 [cs.CY]. URL: https://arxiv.org/abs/2410.04772.

[179] Abeba Birhane et al. " AI auditing: The Broken Bus on the Road to AI Accountability ". In: *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. Los Alamitos, CA, USA: IEEE Computer Society, Apr. 2024, pp. 612–643. DOI: 10.1109/SaTML59370.2024.00037. URL: https://doi.ieeecomputersociety.org/10.1109/SaTML59370.2024.00037.

[180] Miles Brundage et al. *Frontier AI Auditing: Toward Rigorous Third-Party Assessment of Safety and Security Practices at Leading AI Companies*. 2026. arXiv: 2601.11699 [cs.CY]. URL: https://arxiv.org/abs/2601.11699.

[181] Catalina Goanta et al. *The Great Data Standoff: Researchers vs. Platforms Under the Digital Services Act*. 2026. arXiv: 2505.01122 [cs.CY]. URL: https://arxiv.org/abs/2505.01122.

[182] Johnny Tian-Zheng Wei and Robin Jia. *A Strategic Zero Draft: Standards for LLM Memorization*. Tech. rep. Submitted to NIST AI Standards Zero Drafts Pilot Project (https://www.nist.gov/artificial-intelligence/ai-research/nists-ai-standards-zero-drafts-pilot-project-accelerate). National Institute of Standards and Technology (NIST), 2025. URL: https://johntzwei.github.io/pdfs/strategic_zero_draft.pdf.

[183] Jiacheng Liu et al. "Infini-gram: Scaling Unbounded n-gram Language Models to a Trillion Tokens". In: *First Conference on Language Modeling*. 2024. URL: https://openreview.net/forum?id=u2vAyMeLMm.

[184] Tom B. Brown et al. "Language models are few-shot learners". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[185] Jianlin Su et al. "RoFormer: Enhanced transformer with Rotary Position Embedding". In: *Neurocomput.* 568.C (Feb. 2024). ISSN: 0925-2312. DOI: 10.1016/j.neucom.2023.127063. URL: https://doi.org/10.1016/j.neucom.2023.127063.

[186] Noam Shazeer. *GLU Variants Improve Transformer*. 2020. arXiv: 2002.05202 [cs.LG]. URL: https://arxiv.org/abs/2002.05202.

[187] Biao Zhang and Rico Sennrich. "Root mean square layer normalization". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[188] Joshua Ainslie et al. "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 4895–4901. DOI: 10.18653/v1/2023.emnlp-main.298. URL: https://aclanthology.org/2023.emnlp-main.298/.

[189] Alex Andonian et al. *GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch*. Version 2.0.0. Sept. 2023. DOI: 10.5281/zenodo.5879544. URL: https://www.github.com/eleutherai/gpt-neox.

[190] Mohammad Shoeybi et al. "Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism". In: *arXiv preprint arXiv:1909.08053* (2019).

[191] Samyam Rajbhandari et al. *ZeRO: Memory Optimizations Toward Training Trillion Parameter Models*. 2020. arXiv: 1910.02054 [cs.LG]. URL: https://arxiv.org/abs/1910.02054.