Towards More Realistic Extraction Attacks: An Adversarial Perspective

Yash More*
McGill University, Mila, Canada
yash.more@mila.quebec

Prakhar Ganesh*
McGill University, Mila, Canada
prakhar.ganesh@mila.quebec

Golnoosh Farnadi
McGill University, Mila, Canada
farnadig@mila.quebec

Abstract

Language models are prone to memorizing their training data, making them vulnerable to extraction attacks. While existing research often examines isolated setups, such as a single model or a fixed prompt, real-world adversaries have a considerably larger attack surface due to access to models across various sizes and checkpoints, and repeated prompting. In this paper, we revisit extraction attacks from an adversarial perspective—with multi-faceted access to the underlying data. We find significant churn in extraction trends, i.e., even unintuitive changes to the prompt, or targeting smaller models and earlier checkpoints, can extract distinct information. By combining multiple attacks, our adversary doubles ($2\times$) the extraction risks, persisting even under mitigation strategies like data deduplication. We conclude with four case studies, including detecting pre-training data, copyright violations, extracting personally identifiable information, and attacking closed-source models, showing how our more realistic adversary can outperform existing adversaries in the literature.

1 Introduction

Large language models (LLMs) have grown considerably in size (Meta AI, 2024; Zhao et al., 2023), and have become integral to a wide range of tasks such as knowledge retrieval, question answering, code generation, and machine translation.

To complement this growing scale, LLMs are often trained on large amounts of data (Penedo et al., 2024; Soboleva et al., 2023; Gao et al., 2020; Raffel et al., 2020) that may include private information, especially if scraped from the web. As LLMs are prone to memorizing the data they have been trained on, they can be prompted to expose sensitive contexts—making it easier for an adversary to extract information. Naturally, a question arises: How big is the risk imposed due to *memorization*?

Extraction attacks offer an empirical framework to quantify the information leakage in the presence of an adversary. A commonly studied extraction attack is discoverable memorization (Carlini et al., 2023; Kassem et al., 2024), where the adversary extracts targeted information from the training data by prompting the model with a portion of a sentence from the training data to extract the rest. Discoverable memorization has been used in many adversarial settings, including membership inference (Maini et al., 2024), data contamination detection (Ravaut et al., 2024), and copyright violations (Karamolegkou et al., 2023), among others.

Current extraction attacks study memorization trends in LLMs across isolated settings like model sizes, generation hyperparameters, and learning dynamics (Carlini et al., 2023). While effective, they underestimate the risk posed due to a multi-faceted access to the underlying data in the current LLM ecosystem. For instance, we show that an adversary can exploit the sensitivity of LLMs to prompt structure, length, and content, to amplify the information gained. The current accessibility to frequently updated model sizes (Meta AI, 2024); checkpoints (Biderman et al., 2023b; Groeneveld et al., 2024) and a large array of model families such as Llama (Meta AI, 2024), Gemini (Team et al., 2023), and Falcon (Almazrouei et al., 2023), can also create higher extraction risks.

In this paper, we study a more realistic scenario and explore the actual risks posed by composite

* Equal contribution.

Transactions of the Association for Computational Linguistics, vol. 13, pp. 1832–1849, 2025. https://doi.org/10.1162/TACL.a.62
Action Editor: Mohit Iyyer. Submission batch: 5/2025; Revision batch: 7/2025; Published 12/2025.

extraction attacks that can combine information from multiple attacks. More specifically, we ask:

1. **Can adversaries exploit prompt sensitivity and access to multiple checkpoints?** We find that extraction attacks are sensitive to the prompt design, extracting over 20% more data with even minor, unintuitive changes (§5.1). Similarly, we find that an adversary with access to multiple model checkpoints can increase the extraction rates up to $1.5\times$ (§5.2). Thus, an adversary with multi-faceted access can extract far more data than previously observed.

2. **Should the adversary always attack the most vulnerable setup?** Vulnerable setups are also more expensive to attack. Interestingly, we find that on a limited adversarial budget, using composite attacks on less vulnerable but cheaper setups can cause more information leakage (§6.2).

3. **Is data deduplication effective against composite attacks?** We find that data deduplication does reduce memorization, as expected (Carlini et al., 2023). However, adversaries can still exploit the prompt structure and multiple checkpoints to extract more information (§6.4), hence our concerns persist despite deduplication.

4. **How are downstream applications affected by a stronger adversary?** We performed four separate case studies and found that our more realistic adversary improves the p-value of dataset inference in open-source models by up to $2\times$ (§7.1), the extraction of copyright violations by up to 20% (§7.2), the extraction rate of personally identifiable information (PIIs) by $1.5\times$ (§7.3), and training data inference even in closed-source models by 16% (§7.4).

2 Background and Related Work

We first introduce relevant background on extraction attacks in LLMs, followed by an overview of related work on prompt sensitivity and training dynamics in LLMs. Finally, we describe the term *churn* as it applies in our context.

Extraction Attacks in LLMs. Unintended memorization in LLMs can make it prone to information leakage (Tirumala et al., 2022; Carlini et al., 2019; Mattern et al., 2023; Carlini et al., 2022), particularly through extraction attacks (Birch et al., 2023; Carlini et al., 2021, 2023; Nasr et al., 2023). Extraction attacks have gained significant attention in recent years, studied under two primary frameworks: *discoverable memorization* (Carlini et al., 2023; Kassem et al., 2024; Liu et al., 2023b; Biderman et al., 2023a; Tirumala et al., 2022; Huang et al., 2022), where the adversary attempts to extract targeted information, and *extractable memorization* (Nasr et al., 2023; Kandpal et al., 2022; Qi et al., 2024), where the adversary attempts to extract any information about the training data.

We add to the growing body of research on targeted extraction attacks by highlighting the lack of a realistic adversary in the literature. We show the existence of a stronger adversary capable of combining information from various attacks, thus defining *composite discoverable memorization* (§3).

Prompt Sensitivity in LLMs. LLMs are sensitive to changes in their prompts, leading to fluctuations in their performance (Sclar et al., 2024; Liu et al., 2023a). This sensitivity persists across varying model sizes and through fine-tuning and other downstream modifications (Salinas and Morstatter, 2024; Zhu et al., 2023). The sensitivity of prompts can also be misused, and adversarial modifications to prompts can trigger the model to act in unintended ways (Rossi et al., 2024; Liu et al., 2024; Hubinger et al., 2024; Liu et al., 2024).

While several overarching trends studying the impact of prompt design on extraction attacks are present in the literature (Carlini et al., 2023; Kassem et al., 2024; Qi et al., 2024; Tirumala et al., 2022), these trends are often studied in isolation. Motivated by the composability of privacy leakage (McSherry, 2009), we argue that an adversary capable of repeated prompting can combine these trends, and extract more information about the training data than previously reported (§5.1).

Training Dynamics of LLMs. Several recent works have studied the training dynamics of LLMs over time (Tirumala et al., 2022; Liu et al., 2021; Xia et al., 2023). In the context of memorization, Biderman et al. (2023a) explored the impact of model size

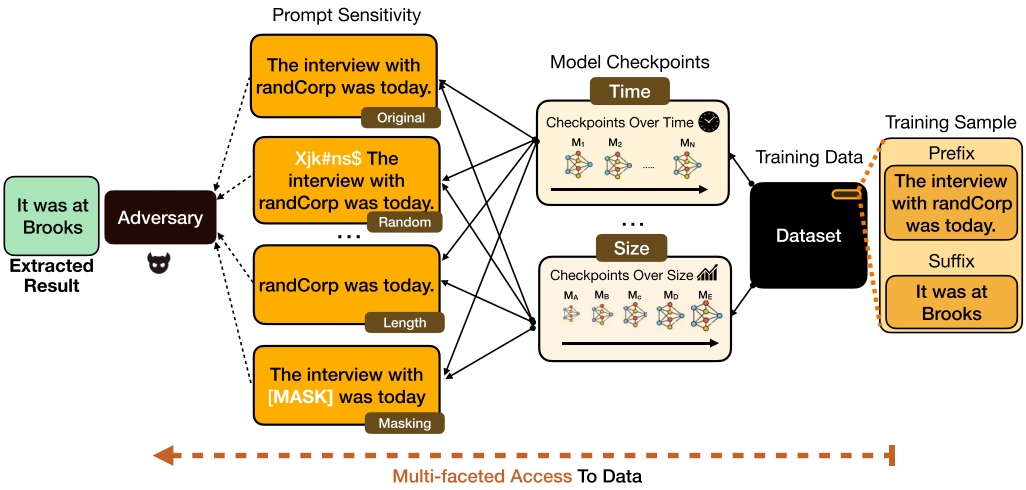

Figure 1: Composability in LLMs. In the real world, an adversary has multi-faceted access to a dataset by (a) exploiting prompt sensitivity, and (b) accessing multiple checkpoints trained on the same data.

and intermediate checkpoints on the dynamics of memorization, revealing a considerable variance in memorized data over time and size. The practice of releasing models in various sizes and regularly updating them can thus increase the attack surface. Similar to prompt sensitivity, we study how adversaries can also exploit access to multiple checkpoints to extract more data (§5.2).

Churn. Churn quantifies the instability of model predictions under updates (Milani Fard et al., 2016; Bahri and Jiang, 2021; Jiang et al., 2021; Adam et al., 2023; Watson-Daniels et al., 2024), i.e., the inconsistency in predictions between a system pre-update vs post-update, by measuring the fraction of examples whose predictions diverge (Milani Fard et al., 2016). While the term churn is traditionally used to describe regressive trends in model predictions, we extend it by highlighting similar regressive trends of extraction attacks under changing prompts and models. Thus, *churn occurs when information not extractable with a stronger setup is instead extractable with weaker setups like shorter prompts, smaller models, or earlier checkpoints.*

3 Re-evaluating Adversarial Strengths

The adversary is central to our work. We start by defining its capabilities and argue that prior work underestimates real-world adversaries. To ensure broad applicability, we assume gray-box access: The adversary can observe model outputs and probabilities but **cannot** access weights, gradients, or even control generation hyperparameters, typical in commercial LLMs. Despite these constraints, we show that adversaries are far more powerful than previously recognized due to their multi-faceted access to the underlying data (see Figure 1).

We focus on discoverable memorization, i.e., we assume access to the ground-truth completion to test whether the extracted information is correct. Here, the adversary is primarily interested in auditing the *memorization* behavior of the model. This is central to many applications, including membership inference, dataset inference, copyright violations, privacy auditing, among others, revisited in §7. That said, as we argue in §8, the larger attack surface and implications of a stronger adversary remain even beyond discoverable memorization.

3.1 Adversary Capabilities

Composability (or self-composability) in privacy (McSherry, 2009) implies that access to multiple outputs from the same data increases the risk of information leakage. Thus, an adversary with multiple access points is much more powerful. In the current landscape of LLMs, such access is not only unsurprising but easily obtainable. Specifically, we focus on two forms of multi-faceted access:

Exploiting Prompt Sensitivity. LLMs are highly sensitive to their input, including its structure and content (Sclar et al., 2024; Liu et al., 2023a; Salinas and Morstatter, 2024; Zhu et al., 2023). While existing studies have focused on improving the prompts for stronger attacks, the

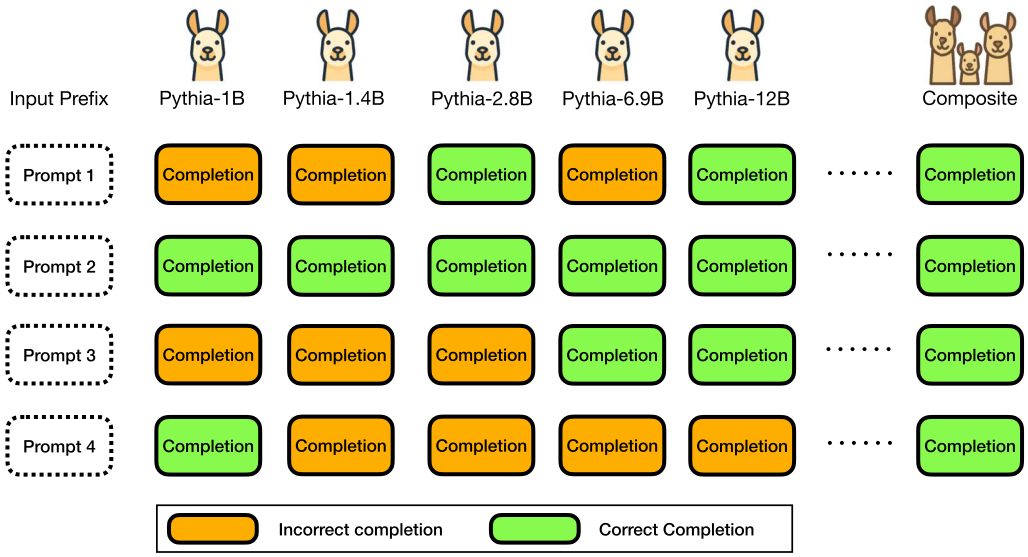

Figure 2: A toy illustration of how churn may emerge across completions from different model sizes. Adversaries can utilize this churn to increase the number of valid extractions.

nuance of prompt sensitivity in LLMs often defies intuitive expectations. For instance, while longer prompts are known to increase the success of extraction attacks (Carlini et al., 2023), our work demonstrates that even shorter prompts can at times exploit vulnerabilities that longer prompts overlook (§5.1).

Given the widespread use of LLMs through both chat interfaces and API calls, restricting model access is not realistic. While most commercial LLMs do have rate limits, they are quite high to be of practical concern. For example, even at the lowest tier subscription of $5, ChatGPT has a 500 query per minute (*qpm*) rate limit for GPT4 and 3500 *qpm* for GPT3.5.[1] Thus, an adversary can prompt millions of generations in just one day, making it easier to exploit prompt sensitivity.

Multiple Checkpoints. LLMs are typically deployed in various sizes to cater to different needs for accuracy and efficiency among users. They also undergo regular updates driven by new data, better learning techniques, evolving security measures, and novel functionalities. Due to the stochastic nature of their training and the impact of scaling, different model sizes or checkpoints might memorize unique portions of the underlying data (Biderman et al., 2023a). Consequently, an adversary with access to various checkpoints across sizes or training can aggregate extracted information (§5.2).

[1] *qpm* stats and subscription rate as of September 2024.

Thus, access to multiple models trained on overlapping datasets substantially increases the attack surface, thereby amplifying the capabilities of adversaries. This level of access has become increasingly common in the current LLM ecosystem. For example, there exist over eight major versions of ChatGPT and ten major versions of Llama, alongside regular minor updates (OpenAI, 2024; Chen et al., 2023). As such, the availability of multiple models with shared training data can significantly increase the risks of information leakage.

3.2 Combining Extraction Attacks

We discussed the elevated risks of multi-faceted access to the training data. Before presenting our empirical results, we first quantify the risks associated with this stronger real-world adversary. We argue that once the adversary gains such broad access, any successful extraction—even if achieved once—renders that specific information vulnerable to the adversary (see Figure 2).

Formally, for a sentence $[\mathbf{p} \parallel \mathbf{x}]$ in the training data, where $\parallel$ represents the concatenation of two strings, the adversary has access to the prompt p and aims to extract information x. Adapting the definition of discoverable memorization from Nasr et al. (2023), we propose:

Definition 3.1 (Composite Discoverable Memorization). For a set of models $\mathbb{G} = \{Gen_i|_{i=1}^{k}\}$,

prompt modifiers $\mathbb{F} = \{F_j|_{j=1}^{r}\}$, and an example $[\mathbf{p} \parallel \mathbf{x}]$ from the training dataset $\mathbb{X}$, we say $\mathbf{x}$ is composite discoverably memorized if $\exists\ Gen_i \in \mathbb{G}$ and $F_j \in \mathbb{F}$, such that, $Gen_i(F_j(\mathbf{p})) = \mathbf{x}$.

$$CDM(\mathbb{G}, \mathbb{F}, \mathbf{p} \parallel \mathbf{x}) = \max_{Gen_i \in \mathbb{G}, F_j \in \mathbb{F}} \mathbb{1}_{Gen_i(F_j(\mathbf{p}))=\mathbf{x}}$$

Prompt modifiers are defined as functions $F_j : \mathcal{W}^* \rightarrow \mathcal{W}^*$ that take a prompt as input and return a modified version of this prompt as output. Here, $\mathcal{W}$ represents a finite set of all tokens in the training data i.e $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$ with w_i representing individual tokens, and $\mathcal{W}^*$ represents the Kleene star operation over $\mathcal{W}$, i.e., a set of all finite length sequences (strings) of tokens in $\mathcal{W}$.

Extraction attacks are often evaluated in the literature using a verbatim match (Carlini et al., 2021, 2023; Nasr et al., 2023; Huang et al., 2022), i.e., the generated text must match the original text perfectly. However, this rigid metric does not take into account the noise in LLM generations, and several works have turned to approximate matching (Qi et al., 2024; Kassem et al., 2024; Liu et al., 2023b; Ippolito et al., 2022). Thus, we also extend our definition of composite extraction attacks to the approximate matching setup as:

Definition 3.2 (Approximate Composite Discoverable Memorization). For a set of models $\mathbb{G} = \{Gen_i|_{i=1}^{k}\}$, prompt modifiers $\mathbb{F} = \{F_j|_{j=1}^{r}\}$, a similarity metric and threshold S, δ, and an example $[\mathbf{p} \parallel \mathbf{x}]$ from the training dataset $\mathbb{X}$, we say $\mathbf{x}$ is approximate composite discoverably memorized if $\exists\ Gen_i \in \mathbb{G}$ and $F_j \in \mathbb{F}$, such that, $S\left(Gen_i\left(F_j\left(\mathbf{p}\right)\right), \mathbf{x}\right) \geq \delta$.

$$ACDM(\mathbb{G}, \mathbb{F}, S, \delta, \mathbf{p} \parallel \mathbf{x}) = \max_{Gen_i \in \mathbb{G}, F_j \in \mathbb{F}} \mathbb{1}_{S\left(Gen_i\left(F_j(\mathbf{p})\right), \mathbf{x}\right) \geq \delta}$$

Here, S is a similarity metric defined as a function $S : (\mathcal{W}^* \times \mathcal{W}^*) \rightarrow [0, 1]$ that takes as input two strings $a, b \in \mathcal{W}^*$, and returns a score between 0 and 1 to represent the similarity between the two input strings, and δ is a threshold that controls the degree of approximate matching. We experiment with various similarity metrics S in §6.3.

4 Experimental Setup

In this section, we outline our central experiment setup to set the stage for our empirical study. Further details about the setup for the case studies (§7) are delegated to their respective sections.

4.1 Models and Dataset

We use the Pythia suite (Biderman et al., 2023b) and OLMo models (Groeneveld et al., 2024) for all our experiments. We focus on the Pythia models throughout the paper, while also providing complementary results on OLMo models to show the generalizability of our analysis. Pythia suite offer access to (a) models of various sizes (1b, 1.4b, 2.8b, 6.9b, and 12b), (b) intermediate checkpoints during training (a total of 154 checkpoints, with 144 of them equally spaced, i.e., at every $1k$ training steps), and (c) the training data (Pile dataset [Gao et al., 2020]) as well as the training order, which is the same for all model sizes. This level of accessibility and control over the training setup allows us to simulate the real-world availability of models across sizes and different checkpoints over time.

OLMo models were trained on the Dolma dataset (Soldaini et al., 2024) and also offer access to (a) intermediate model checkpoints during training, and (b) the complete training data order.

4.2 Evaluation Methodology

We now describe our evaluation methodology. Similar to Carlini et al. (2023), we sample a representative portion of the training data for analyzing the performance of our extraction attacks. More specifically, we uniformly sample 100,000 sequences (prompts) from the first $100k$ steps (batches) of the training data for Pythia. This sampling strategy is important because we choose model checkpoints for evaluation starting at step $100k$, which ensures that every sentence evaluated for memorization has been seen by each checkpoint under consideration (as illustrated in Figure 3).

Each sequence sampled is exactly 2049 tokens. For our analysis, we employ a consistent method of partitioning each sequence into a prompt and completion at the midpoint, i.e., 1024 tokens. Formally, for a sentence $s_{1:2049}$, prompt length l_p, and completion length l_x, the example $[\mathbf{p} \parallel \mathbf{x}]$ is defined as $\mathbf{p} = s_{1024-l_p:1024}$ and $\mathbf{x} = s_{1024:1024+l_x}$. This partitioning allows us to systematically vary

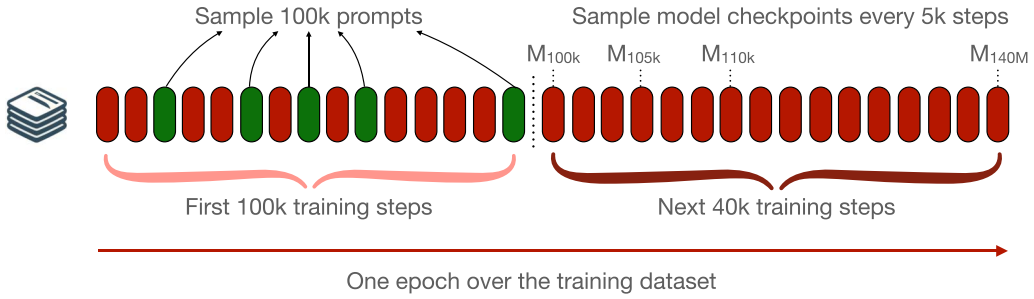

Figure 3: Choosing prompts (pre $100k$ steps) and checkpoints (post $100k$ steps) for evaluation of Pythia.

the prompt length and design while comparing the same completion, and vice versa. We use the same approach for OLMo, with the training step $300k$ (instead of $100k$) being the cut-off point.

For the Pythia suite, unless otherwise specified, we use a prompt length of $l_p = 50$, a completion length of $l_x = 50$, the Pythia-6.9b model, and the 140k training step checkpoint, evaluating the extraction attacks using verbatim match. For OLMo, we use the OLMo-7b model and the 500k training step checkpoint as defaults, while the rest of the configuration is the same as Pythia.

5 Churn in Extraction Trends

Churn (Milani Fard et al., 2016), as previously introduced in §2, refers to regressive variance for individual extracted information despite an overall improvement in the extraction rates. For instance, although using a longer prompt is often associated with stronger extraction rates (Carlini et al., 2023; Biderman et al., 2023a), we observe trends that exhibit churn, i.e., certain information is instead extractable only with shorter prompts but not with longer prompts. These non-monotonic and locally regressive trends of certain sentences (i.e., churn) can be exploited by an adversary with multifaceted access to the data to execute a composite extraction attack. We study the factors that may lead to *churn* such as (a) prompt sensitivity, and (b) access to models of varying sizes and training checkpoints.

5.1 Prompt Sensitivity

We start by examining prompt sensitivity, focusing on how trends in prompt design can lead to churn.

Prompt Length. Prompt length is a commonly studied parameter in extraction attacks, and it has been shown that longer prompts lead to better extraction (Carlini et al., 2023). This is intuitive, as conditioning the model with more text from training would increase the likelihood of extraction.

However, we show in Figure 4(a) that the composite extraction rate (Definition 3.1) across varying prompt lengths is noticeably higher than the extraction rate at even the largest prompt length at 500 tokens. Thus, certain information extractable with shorter prompts remains elusive even with the longest prompt, due to prompt sensitivity in LLMs, as discussed in §3.1. Consequently, an adversary can exploit this churn across the prompt length to extract more information. We see similar trends for OLMo in Figure 5.

Prompt Structure. Next, we explore the structure of prompts to identify where churn can emerge. We introduce noise into the prompts by masking and removing random tokens (Figure 4(d)); and as a prefix in the form of random numeric and alphanumeric strings (Figure 4(e)). Despite introducing only a small amount of noise, we observe a significant drop in extraction rates. This indicates that the contiguous prompt from the training data is crucial for extracting information, and any disruption to this prompt can significantly hurt its capabilities.

Yet, we do see churn in extraction trends, with a larger impact of the noisy prefixes. This further highlights how an adversary can exploit repeated prompting, even with seemingly unintuitive changes like masking or removing random tokens and adding a random prefix to the input prompt.

Note that the churn trends under prompt sensitivity, both for prompt structure and prompt length, highlight the increased extraction risks without access to new information. For instance, if an adversary has access to the prompt of length

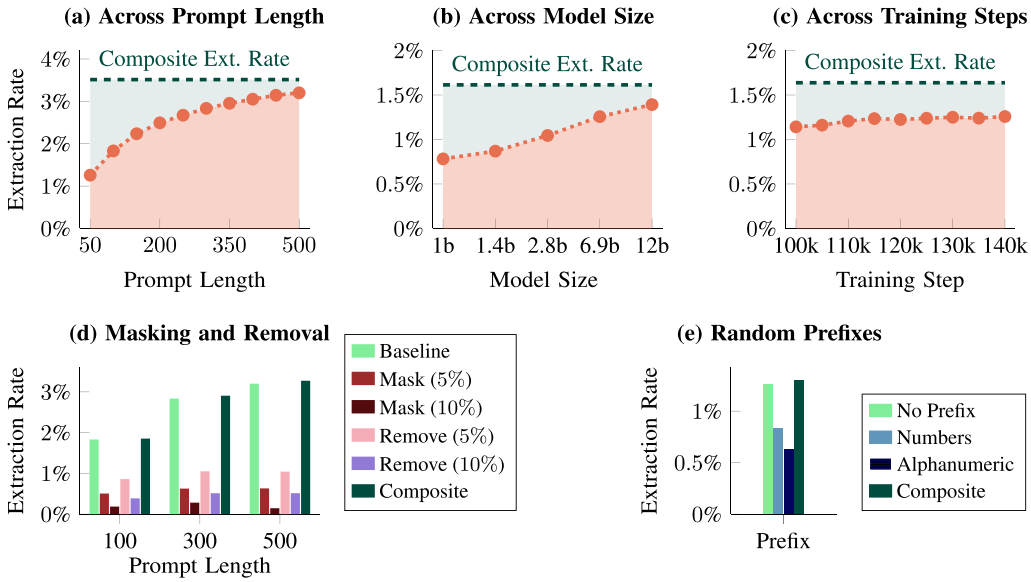

Figure 4: Extraction rates under prompt sensitivity and across models for Pythia. **(a)** Increasing prompt length improves extraction, with the composite extraction rate better than even at prompt length 500. **(b, c)** Increasing model size and training steps show similar trends, with the largest impact of the composite extraction rate seen in training steps, increasing the extraction rate $1.5\times$ compared to a single checkpoint. **(d)** Randomly masking or removing tokens severely hurts the extraction rate, highlighting the importance of prompt structure. **(e)** Adding a random prefix can also contribute to the composite extraction rate.

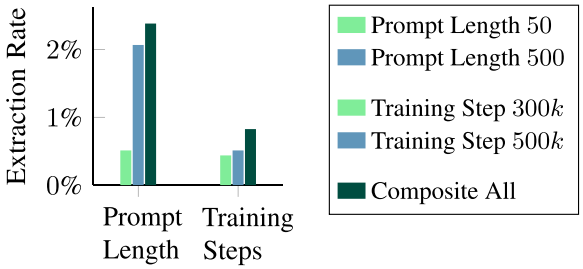

Figure 5: Composite extraction attack results across 10 prompt lengths (same as Pythia) and 11 training steps (equidistant between $300k$ and $500k$), compared against isolated setups, for OLMo.

500 tokens, they can expand the attack surface and thereby the extraction rate simply by removing parts of the prompt (prompt length composite attack from Figure 4(a)), adding noise (prompt structure composite attack from Figure 4(d, e)), etc., without needing any additional knowledge.

One might argue that as the number of prompt variations increases, every sentence could become extractable. However, that is not true; not all sentences are extractable. Yin et al. (2024) showed that knowledge not present in an LLM will not be extractable even after prompt optimization, while Schwarzschild et al. (2024) also showed similar trends when attempting to extract a given completion. Consequently, prompting an LLM to regurgitate certain sentences, even with various prompt modifications, demonstrates a genuine extraction risk and underscores the extent of memorization in LLMs (Carlini et al., 2021).

5.2 Multiple Checkpoints

Model Size. The model size has long been known to influence learning trends, and our results in Figure 4(b) reflect this phenomenon. Larger models tend to memorize more information, which makes them more vulnerable to extraction attacks. However, our results also indicate that the composite extraction rate is higher than the extraction rate of any single model, highlighting the churn in these trends. Biderman et al. (2023a) also conducted an empirical study on the overlap between memorized data across model sizes and found that up to 10% data memorized by smaller models is not memorized by larger models. Combining our insights with existing literature, it's clear that releasing models in different sizes increases the extraction risks.

Model Updates. We also analyze model updates over time using intermediate checkpoints in Figure 4(c), where we observe the most significant churn in our study. Unsurprisingly, attacking

Figure 6: Real examples of churn. **(Left)** Prompts of different lengths can contribute uniquely to the model extraction, and longer contexts aren't always better. **(Right)** Similarly, different model sizes contribute uniquely to the composite extraction rate, showing the importance of churn across model size.

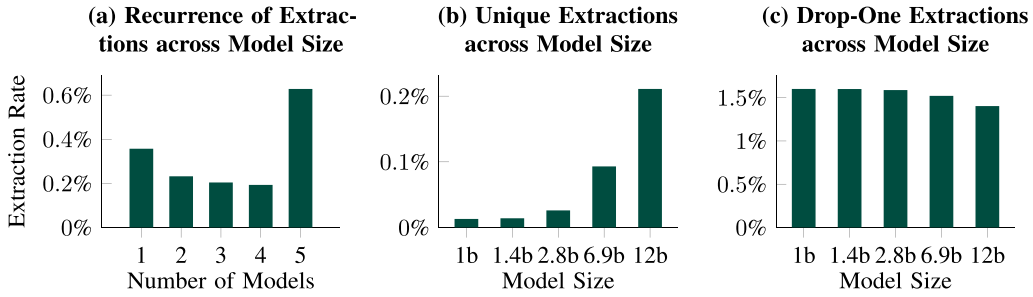

Figure 7: Granular trends of churn across model size for Pythia. **(a)** Number of models that successfully extract the same information, i.e., whether only 1 out of 5 models extracts a given sample, or whether 2, 3, 4, or all 5 do. (The *x*-axis here indicates the exact number of models that extract the same samples.) Even though the majority of samples are extractable by all 5 models, a significant amount of extractions are unique to a subset, or even a single model. **(b)** While larger models contribute more unique extractions, each model, regardless of size, provides some unique extractions. **(c)** Composite extraction rates after dropping a single model (*x*-axis: dropped model, *y*-axis: resulting extraction rate) show that extraction rates remain stable even if we remove the contributions of the biggest model.

models at later stages of training is more successful, as seen in the literature (Tirumala et al., 2022; Biderman et al., 2023a; Jagielski et al., 2023). But remarkably, the churn here is significantly powerful and by exploiting composability, an adversary can increase their extraction rate by more than $1.5\times$. This underscores the impact of stochasticity in model training on extraction attacks and reveals that regular model updates, typically considered beneficial in the LLM ecosystem, create a powerful adversary. We also see similar results for OLMo in Figure 5.

We provide some examples of extractions from the Pile dataset that show regressive trends, i.e., successful extraction using weaker setups, and highlight the value of churn, in Figure 6.

5.3 Unique Extractions

Next, we study the contribution of each setup to the composite extraction rate. We focus on trends across model size, while providing results for other axes in the appendix (§A). We find that while a majority of extracted samples are extractable with all 5 model sizes, there is a significant portion of extractions unique to a few (or even just one) model(s) (Figure 7(a)). Examining the distribution of extractions unique to individual models (Figure 7(b)), we observe that Pythia-6.9b produces a substantial number of unique extractions not found even in the larger Pythia-12b. Smaller models, such as Pythia-1b, 1.4b, and 2.8b, also contribute non-trivially.

Finally, we study composite extraction rates when all models except one are used (Figure 7(c)), to quantify the contribution of each model. We find that even after dropping the largest model, the remaining models achieve high composite extraction rates, indicating that no single model is essential for strong extraction. Given that attacking the most vulnerable model can be expensive, this shows that an adversary can take advantage of

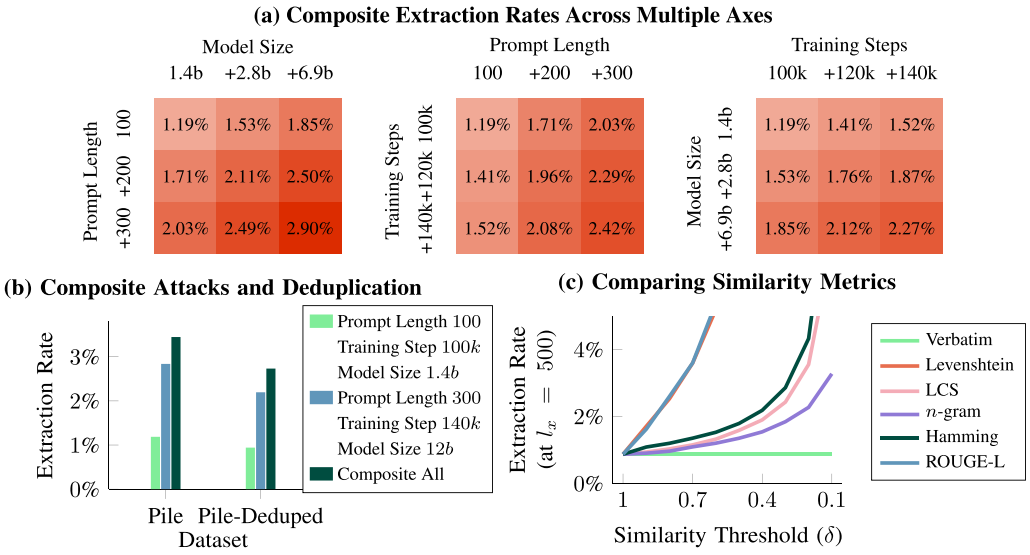

Figure 8: Towards more realistic extraction by combining various churn trends and with approximate matching. **(a)** Combining two axes at a time, we see a monotonically increasing trend in extraction rates as we gain more points of access to the underlying data, highlighting the growing power of the adversary. **(b)** Combining all axes of attack results in a significant increase in extraction rate for both standard and deduplicated setups. **(c)** Various similarity metrics have distinct trends as we decrease the threshold and allow for looser approximations, thus the choice of similarity metric depends on the context of the attack.

the churn to instead perform a composite attack on less vulnerable but cheaper models (more details in §6.1).

6 Towards Realistic Extraction Attacks

With a better understanding of the churn, we now evaluate a more realistic measure of leakage in extraction attacks, by investigating (a) composability across multiple axes, (b) cost of composite attacks, (c) approximate matching, and (d) deduplication.

6.1 Combining Multiple Axes of Churn

In the previous section, we saw how churn can impact individual axes of variability. However, a real-world adversary can take advantage of all factors simultaneously, thus significantly increasing their extraction rates. We start by analyzing two axes at a time in Figure 8(a). For all pairs of variability, the overall composite extraction rate (bottom right) is $2-3\times$ higher than the base setup (top left) and $1.5-2\times$ higher than the composite extraction rates along one axis (top right and bottom left). Furthermore, when all three axes are combined, depicted in Figure 8(b), the extraction rates grow even higher, albeit with diminishing gains. Thus, a real-world adversary can extract far more training data than shown in existing literature.

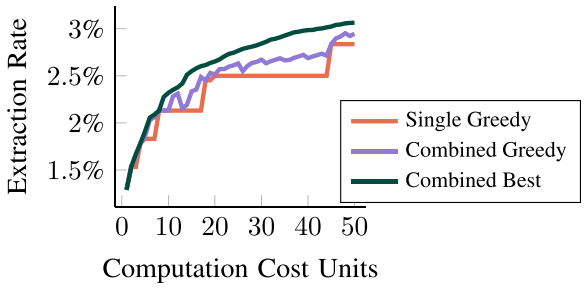

Figure 9: Various strategies of utilizing the adversarial budget and the resulting extraction rates.

6.2 Computation Cost of Composite Attacks

We now study the computational cost of performing composite extraction attacks. We define our cost in units relative to attacking the Pythia-1.4b model with a prompt length of 100 (the cheapest setup to attack in our multiple axes setting). Cost scales linearly with model size (i.e., attacking the Pythia-2.8b model with prompt length 100 costs 2 units) and quadratically with prompt length (i.e., attacking the Pythia-1.4b model with prompt length 200 costs 4 units), while remaining constant across different checkpoints. The resulting extraction trends for all axes of churn combined under varying cost budgets are shown in Figure 9.

We study three different strategies for utilizing the given resources. First, we consider an

adversary that greedily attacks the most expensive setup, i.e., prioritizing the largest prompt length, model size, and latest checkpoint, while leaving any leftover resources unused. The extraction rate stays flat until the budget can support a more expensive setup, causing sharp jumps and creating a staircase pattern. This strategy mirrors prior work, where each setup is evaluated in isolation. Next, we extend the previous strategy to utilize the unused resources, and instead of attacking only a single setup, the adversary now selects multiple setups, still greedily choosing to fit the remaining budget.

Finally, we define the third strategy, which is the most effective: searching all combinations to maximize the composite extraction rate, rather than greedily targeting the most vulnerable setups. This consistently outperforms the staircase trend even at the jumps, i.e., even when the greedy strategy utilizes all available budget, showing that combining extractions from less vulnerable setups can exceed the returns of attacking the most expensive setup.

6.3 Approximate Matching

As discussed in §3.2, evaluating extraction attacks under verbatim match can underestimate the true risk of extraction. Here, we analyze approximate composite discoverable memorization (Definition 3.2) across various similarity metrics S to examine their behavior under changing δ in Figure 8(c). Solely for this discussion, we increase the completion length $l_x = 500$, to allow for meaningful extraction even with approximate matching.

Our results reveal intriguing trends. First, we study evaluations based on the Levenshtein ratio metric and observe that even the strict threshold of $\delta = 0.95$ doubles the extraction attack rate compared to a verbatim match. This threshold signifies a minimum 95% overlap between generated and original text. Clearly, we miss out on a considerable amount of leaked information by relying only on verbatim matches. As δ decreases, the extraction rate increases exponentially, as the Levenshtein ratio becomes less reliable under looser constraints. We also see similar trends for ROUGE-L scores.

For similarity metrics like longest common substring (LCS), Hamming distance, and n-gram matching, even lower values of similarity (δ) can contribute meaningfully to extraction attacks. Unsurprisingly, we observe patterns of increasing extraction rates as before, albeit slower. The diverse trends underscore the choice of the approximation metric as highly context-dependent. A more thorough examination of which metrics best serve particular applications is left for future work.

6.4 Data Deduplication

A commonly recommended solution to memorization is data deduplication, involving the removal of duplicate data entries within a dataset (Carlini et al., 2023). While costly, data deduplication represents a critical aspect of data curation and has been shown to mitigate extraction risks (Carlini et al., 2023). To understand the role of data deduplication in our discussion, we repeat our experiments using Pythia models trained on the deduplicated Pile dataset. The results are collected in Figure 8(b).

In line with existing literature, data deduplication reduces the extraction rate. Interestingly, however, we observe persistent trends: the presence of a stronger adversary due to multi-faceted dataset access. Thus, while beneficial, data deduplication does not alter our fundamental conclusions; real-world adversaries with multi-faceted access to the underlying data can extract substantial information even post-deduplication. Future work on incorporating more concrete frameworks like differential privacy is needed, to better understand such adversaries, particularly from the perspective of privacy protection under multi-access systems.

7 Case Studies with Stronger Adversary

We conclude by highlighting the value of our stronger adversary through various case studies.

7.1 Detecting Pre-Training Data

Extraction attacks identify whether certain data was included in a model's training set. This can be valuable in assessing whether a model is trained on proprietary or sensitive data without permission, evaluating data contamination and leakage in various benchmarks, ensuring regulatory compliance with data governance policies, or even academic research to track data contamination.

While membership inference attacks (MIAs) have been used to detect pre-training data, Maini

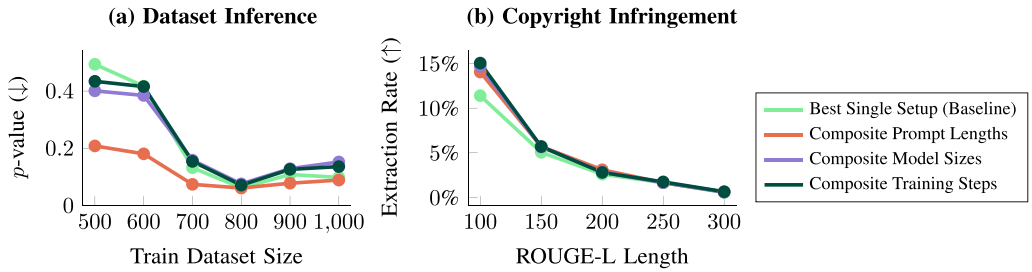

Figure 10: **(a)** p-value for dataset inference (lower is better) across different dataset sizes. The results show significant improvement under the composite setup across different prompt lengths. **(b)** Extraction rate for different ROUGE-L length thresholds, marking potential copyright violations generated by the LLM. Extraction rates with composite setups are consistently higher than the baseline setup.

et al. (2024) argue that in reality, MIAs are as good as random guessing. They show that these attacks learn how to distinguish between *concepts*, and not actual text, highlighting the importance of using IID data of members and non-members to appropriately perform dataset inference.

We borrow their setup and extend it to the composite setting by increasing the size of the training set for learning correlations. Thus, our composite setting can be alternatively seen as an augmentation technique. We record the p-value of the null hypothesis *''the dataset was not used for training''* for the Pile dataset in Figure 10(a), under different sizes of the original training data.

We find that p-values for the composite setting across prompt lengths are noticeably lower than the baseline, especially at smaller dataset sizes. Thus, our adversary requires less data to get the same p-value. The dataset inference setup by Maini et al. (2024) requires obtaining IID data, which can be difficult to find. Hence, reducing the amount of such data required can be extremely useful. Interestingly, we did not find strong composition trends across model checkpoints, possibly because membership information can change drastically across models, and thus, combining information from multiple checkpoints might not be helpful.

7.2 Copyright Infringement

Copyright issues due to LLMs regurgitating their training data have been heavily studied in recent literature. Karamolegkou et al. (2023) discuss different thresholds for quoting a text ad verbatim that can be considered a violation of fair use, for example, a 50-word threshold for magazine articles, chapters, etc., while a 300 word threshold for books. The authors suggest using ROUGE-L lengths (longest common subsequence) as a measure of text reproduction and potential violations.

We plot the distribution of ROUGE-L lengths for 2,000 randomly chosen examples in Figure 10(b), comparing the strongest baseline and the composite settings. We find that our adversary produces more potential copyright violations than the baseline, highlighting an underestimation of such risks in existing literature. While copyright law is complex and extraction alone may not imply a violation, our focus is on strengthening the technical underpinnings of copyright issues in LLMs.

7.3 PII Extraction Risk

Another commonly studied risk of memorization is extracting personally identifiable information (PIIs). We use the setup of Li et al. (2024) to create our PII extraction test set from the Pile dataset. We use GLiNER (Zaratiana et al., 2024) to detect 2000 unique PIIs in the Pile dataset, followed by cutting the sentence right before the PII to create the input prompt. These prompts were fed to the model, and the attack is considered successful if the correct PII is generated anywhere within the first 100 tokens, marking the risk of PII leakage (Li et al., 2024).

We record the extraction risk for the best single setup and composite extraction risks across model checkpoints and model sizes. Since the prompts in this setup are of varying lengths, we do not extend our changing prompt lengths setting to this case study. Similar to Definition 3.1, the composite PII extraction is considered successful if the PII is present in the generation of at least one of the models. The results are collected in Table 1, and continuing previous trends, we see a noticeable

Setup	Extraction Rate
Best Single Setup	22.16%
Composite Model Sizes	30.97%
Composite Training Steps	33.07%

Table 1: Composite attacks for PII extraction.

increase in the extraction rate for an adversary with access to multiple checkpoints.

7.4 Closed-Source Models

Our study till now has focused on open-source models due to the availability of training data for verifying the success of our attacks. However, recent work by Duarte et al. (2024) has shown how similar studies can be performed even when a model's training data is undisclosed.

Duarte et al. (2024) propose DE-COP, a method for detecting memorization in LLMs using a multiple-choice task inspired by counterfactual memorization. The approach presents the target model with four options: one original text passage and three paraphrases generated using a different LLM, in this case, Claude 2. Models tend to choose the original text more frequently if seen during training, thus signaling memorization.

The authors validate this method with a novel benchmark—BookTection—which consists of excerpts from 165 books published both before and after 2023. Since closed-source models released in 2022 could not have been trained on books published after 2023, this serves as a reliable non-member set. The books pre-2023 form the member-set. The method compares a model's performance on each book to a baseline performance (computed from non-member books), using it to assess whether a specific book was part of the model's training data. Each book can have multiple passages, so the authors combine the accuracy per passage for the detection of the book.

For our study, we focus on the experiments on LLaMA-2 70B using the BookTection dataset with varying passage lengths. The authors found that shorter passages yield higher F1 scores: 64, 128, and 256-token passages achieve F1 scores of 0.67, 0.65, and 0.64, respectively. Building on this, we extend their method to a composite setting that combines different lengths. This composite significantly boosts detection performance on LLaMA-2 70B, achieving an F1 score of 0.78, underscoring the generalizability of our approach.

8 Discussion

By highlighting the multi-faceted access adversaries have in the current LLM landscape, our work shows that existing literature greatly underestimates information leakage risks, thus emphasizing the importance of explicitly considering the adversarial perspective and the composability of information leakage in extraction attacks. Our work provides a foundation for future exploration and defense against more realistic extraction attacks, contributing to a secure and robust management of the risks associated with memorization in LLMs.

Potential Defenses. We studied the threats posed by real-world adversaries and showed that existing defence methods (such as data deduplication), while undoubtedly useful, are prone to the same risks of composability. However, we did not propose defense techniques to deal with this adversary.

Firstly, it's important to recognize that not all instances of discoverable memorization are harmful, and therefore may not require a defense. For instance, defending against improved methods of detecting copyright violations or data contamination is inadvisable. Such efforts could hinder those seeking to determine whether their data was misused by companies or developers.

However, when we do want to defend against these attacks in more harmful scenarios, future research could include ways to disrupt multi-faceted access to the dataset. For example, shuffling and re-chunking training sequences for each model can break the link between specific sequences and model behavior. Since many existing LLM attacks operate at the sequence level (Meeus et al., 2024), this simple randomization can significantly increase the difficulty of combining information across checkpoints or model sizes.

Other defense strategies, such as anonymizing sensitive data before training (Yu et al., 2024) or applying differential privacy (DP) during training (Yu et al., 2022), can also help. However, their effectiveness may decline against a stronger adversary. Finally, beyond modifying training, one may wish to defend already trained models against such attacks. While this is challenging, several practical strategies could be potentially helpful. Output perturbation techniques, such as adding noise or rephrasing responses, can reduce

information leakage, even though a determined adversary may still bypass them. Access control measures, such as rate limiting and monitoring for suspicious prompts, also offer practical defenses in real-world deployments.

Beyond Discoverable Memorization. Our analysis focuses on the risks posed by extraction attacks under the lens of discoverable memorization. However, our arguments on the increased privacy surface apply to any adversary with multi-faceted access to the underlying data. Moreover, our experiments on Pythia and OLMo represent a controlled setup where the underlying models were trained on the exact same data and data order. However, in reality, multiple models from the same family might have some differences in their training. Hence, translating our findings to other forms of privacy attacks and dataset homogenization in the real world is an important direction for future research.

Acknowledgments

We thank the anonymous reviewers and the action editor Kai-Wei Chang from TACL, for their continued feedback and comments that helped improve our work.

Funding support for project activities has been partially provided by Canada CIFAR AI Chair, Google award, MITACS, FRQNT, and NSERC Discovery Grants program. We also express our gratitude to Compute Canada for their support in providing facilities for our evaluations.

References

George Adam, Benjamin Haibe-Kains, and Anna Goldenberg. 2023. Maintaining stability and plasticity for predictive churn reduction. *arXiv preprint arXiv:2305.04135*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Dara Bahri and Heinrich Jiang. 2021. Locally adaptive label smoothing improves predictive churn. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 532–542. PMLR.

Stella Biderman, USVSN Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. 2023a. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023b. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Lewis Birch, William Hackett, Stefan Trawicki, Neeraj Suri, and Peter Garraghan. 2023. Model leeching: An extraction attack targeting LLMs.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy*. IEEE. https://doi.org/10.1109/SP46214.2022.9833649

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, pages 267–284, USA. USENIX Association.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is chatGPT's behavior changing over time? *arXiv preprint arXiv:2307.09009*. https://doi.org/10.1162/99608f92.5317da47

André V. Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. 2024. De-cop: Detecting copyrighted content in language models training data.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.841

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047. https://doi.org/10.18653/v1/2022.findings-emnlp.148

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. 2024. Sleeper agents: Training deceptive llms that persist through safety training.

Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. 2022. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*. https://doi.org/10.18653/v1/2023.inlg-main.3

Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Guha Thakurta, Nicolas Papernot, et al. 2023. Measuring forgetting of memorized training examples. In *The Eleventh International Conference on Learning Representations*.

Heinrich Jiang, Harikrishna Narasimhan, Dara Bahri, Andrew Cotter, and Afshin Rostamizadeh. 2021. Churn reduction via distillation. In *International Conference on Learning Representations*.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. 2023. Copyright violations and large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7412. https://doi.org/10.18653/v1/2023.emnlp-main.458

Aly M. Kassem, Omar Mahmoud, Niloofar Mireshghallah, Hyunwoo Kim, Yulia Tsvetkov, Yejin Choi, Sherif Saad, and Santu Rana. 2024. Alpaca against vicuna: Using LLMs to uncover memorization of LLMs. *arXiv preprint*

arXiv:2403.04801. https://doi.org/10.18653/v1/2025.naacl-long.421

Qinbin Li, Junyuan Hong, Chulin Xie, Jeffrey Tan, Rachel Xin, Junyi Hou, Xavier Yin, Zhun Wang, Dan Hendrycks, Zhangyang Wang, Bo Li, Bingsheng He, and Dawn Song. 2024. LLM-PBE: Assessing data privacy in large language models. *Proceedings of the VLDB Endowment*, 17(11):3201–3214. https://doi.org/10.14778/3681954.3681994

Leo Z. Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. 2021. Probing across time: What does roBERTa know and when? *CoRR*, abs/2104.07885.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9). https://doi.org/10.1145/3560815

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023b. Trustworthy LLMs: A survey and guideline for evaluating large language models' alignment. In *Socially Responsible Language Modelling Research*.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, Kailong Wang, and Yang Liu. 2024. Jailbreaking chatgpt via prompt engineering: An empirical study. https://doi.org/10.1145/3663530.3665021

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024. LLM dataset inference: Did you train on my dataset? *arXiv preprint arXiv:2406.06443*.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343. https://doi.org/10.18653/v1/2023.findings-acl.719

Frank D. McSherry. 2009. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30. https://doi.org/10.1145/1559845.1559850

Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. 2024. Sok: Membership inference attacks on llms are rushing nowhere (and how to fix it). *arXiv preprint arXiv:2406.17975*. https://doi.org/10.1109/SaTML64287.2025.00028

Meta AI Meta AI. 2024. Introducing meta Llama 3: The most capable openly available LLM to date.

Mahdi Milani Fard, Quentin Cormier, Kevin Canini, and Maya Gupta. 2016. Launch and iterate: Reducing prediction churn. *Advances in Neural Information Processing Systems*, 29.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models.

OpenAI. 2024. ChatGPT Documentation: Models.

Guilherme Penedo, Hynek Kydlíček, Leandro von Werra, and Thomas Wolf. 2024. Fineweb.

Zhenting Qi, Hanlin Zhang, Eric Xing, Sham Kakade, and Himabindu Lakkaraju. 2024. Follow my instruction and spill the beans: Scalable data extraction from retrieval-augmented generation systems. *arXiv preprint arXiv:2402.17840*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Mathieu Ravaut, Bosheng Ding, Fangkai Jiao, Hailin Chen, Xingxuan Li, Ruochen Zhao, Chengwei Qin, Caiming Xiong, and Shafiq Joty. 2024. How much are large language models contaminated? A comprehensive survey and the LLMsanitize library. *arXiv preprint arXiv:2404.00699*.

Sippo Rossi, Alisia Marianne Michel, Raghava Rao Mukkamala, and Jason Bennett Thatcher.

2024. An early categorization of prompt injection attacks on large language models.

Abel Salinas and Fred Morstatter. 2024. The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance. https://doi.org/10.18653/v1/2024.findings-acl.275

Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C. Lipton, and J. Zico Kolter. 2024. Rethinking LLM memorization through the lens of adversarial compression.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R. Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.840

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, et al. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290.

Jamelle Watson-Daniels, Flavio du Pin Calmon, Alexander D'Amour, Carol Long, David C. Parkes, and Berk Ustun. 2024. Predictive churn with the set of good models. *arxiv preprint arXiv:2402.07745*.

Mengzhou Xia, Mikel Artetxe, Chunting Zhou, Xi Victoria Lin, Ramakanth Pasunuru, Danqi Chen, Luke Zettlemoyer, and Ves Stoyanov. 2023. Training trajectories of language models across scales. In *61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, pages 13711–13738. Association for Computational Linguistics

(ACL). https://doi.org/10.18653/v1/2023.acl-long.767

Xunjian Yin, Xu Zhang, Jie Ruan, and Xiaojun Wan. 2024. Benchmarking knowledge boundary for large language model: A different perspective on model evaluation. *arXiv preprint arXiv:2402.11493*.

Da Yu, Sivakanth Gopi, Janardhan Kulkarni, Zinan Lin, Saurabh Naik, Tomasz Lukasz Religa, Jian Yin, and Huishuai Zhang. 2024. Selective pre-training for private fine-tuning. *Transactions on Machine Learning Research*.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. 2022. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. Gliner: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376. https://doi.org/10.18653/v1/2024.naacl-long.300

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Zhenqiang Gong, and Xing Xie. 2023. PromptBench: Towards evaluating the robustness of large language models on adversarial prompts. https://doi.org/10.1145/3689217.3690621

A Unique Extraction Results

We provide additional results on the trends of churn across model checkpoints (Figure 11) and prompt lengths (Figure 12). We find similar trends as in the main paper.

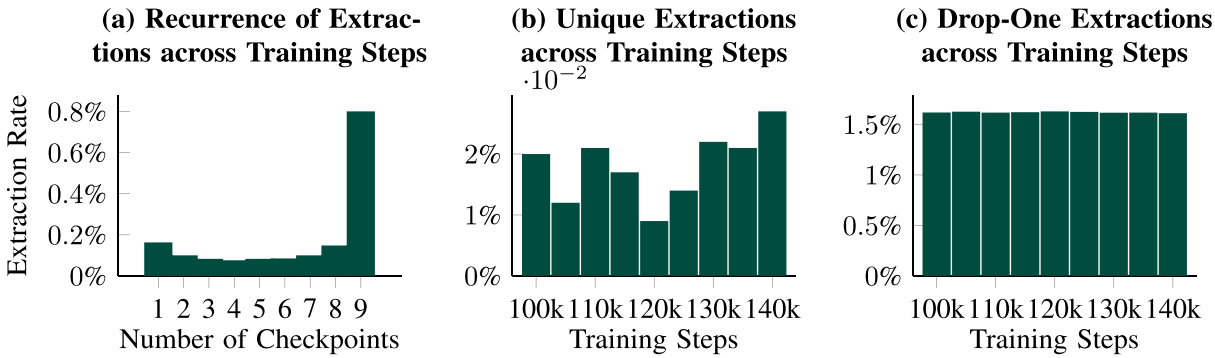

Figure 11: Granular trends of churn across training steps for Pythia. **(a)** Number of model checkpoints extracting the same information. Even though the majority of samples are extractable by all 9 checkpoints, a significant amount of extractions are unique to a subset, or even a single model. **(b)** While later model checkpoints contribute more unique extractions, each model, regardless of size, provides significant unique extractions. **(c)** Composite extraction rates after dropping a single model checkpoint (*x*-axis: dropped checkpoint, *y*-axis: resulting extraction rate) show that extraction rates remain stable even if we remove the contributions of the last checkpoint.

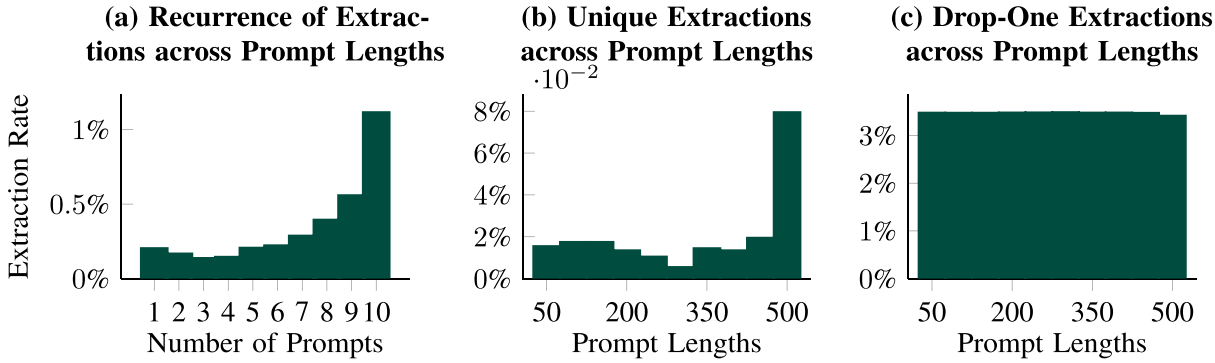

Figure 12: Granular trends of churn across prompt lengths for Pythia. **(a)** Number of prompts extracting the same information. Even though the majority of samples are extractable by all 10 prompt lengths, a significant amount of extractions are unique to a subset, or even a single prompt length. **(b)** While longer prompt lengths contribute more unique extractions, each prompt length provides some unique extractions. **(c)** Composite extraction rates after dropping a single prompt length (*x*-axis: dropped prompt length, *y*-axis: resulting extraction rate) show that extraction rates remain stable even if we remove the contributions of the longest prompt length.