Privacy Ripple Effects from Adding or Removing Personal Information in Language Model Training

Jaydeep Borkar[1], Matthew Jagielski[2], Katherine Lee[2], Niloofar Mireshghallah[3], David A. Smith[*1], Christopher A. Choquette-Choo[*2]

[1]Northeastern University, [2]Google DeepMind, [3]University of Washington

Correspondence: borkar.j@northeastern.edu and cchoquette@google.com

Abstract

Due to the sensitive nature of personally identifiable information (PII), its owners may have the authority to control its inclusion or request its removal from large-language model (LLM) training. Beyond this, PII may be added or removed from training datasets due to evolving dataset curation techniques, because they were newly scraped for retraining, or because they were included in a new downstream fine-tuning stage. We find that the amount and ease of PII memorization is a dynamic property of a model that evolves throughout training pipelines and depends on commonly altered design choices. We characterize three such novel phenomena: (1) similar-appearing PII seen later in training can elicit memorization of earlier-seen sequences in what we call *assisted memorization*, and this is a significant factor (in our settings, up to 1/3); (2) adding PII can increase memorization of other PII significantly (in our settings, as much as $\approx 7.5\times$); and (3) removing PII can lead to other PII being memorized. Model creators should consider these first- and second-order privacy risks when training models to avoid the risk of new PII regurgitation.

1 Introduction

One of the most common methods to adapt large language models like ChatGPT (Achiam et al., 2023) and Gemini (Gemini Team et al., 2023) for specific applications is to fine-tune them on domain-specific datasets.[1] When these datasets contain private or personal data, models may be at risk of memorizing[2] and regurgitating (Carlini et al., 2022b) this information. Though it is common to filter out sensitive information[3] such as PII (Gemma Team et al., 2024b), some sensitive information may still remain (Vakili et al., 2022). Moreover, some downstream tasks, such as healthcare, may require PII, making eliminating PII completely from model training datasets challenging.

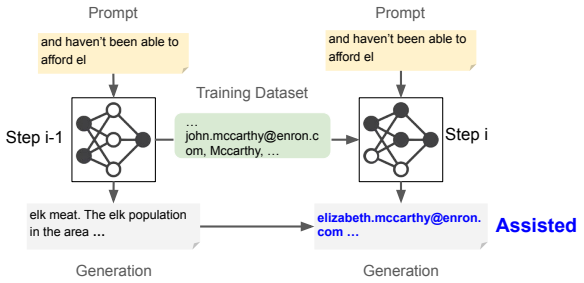

Figure 1: We explore a phenomenon we call *assisted memorization*, where unique PII that appeared earlier in the training at step $i-1$ and was not extracted at that step becomes extractable at a later step i, after fine-tuning on *other* PII.

Modern-day language models deployed in real-world settings are also increasingly dynamic: it is common practice to continually update or retrain them with new and/or additional data (Razdaibiedina et al., 2023; Ke et al., 2023; Jang et al., 2022; Jin et al., 2022), e.g., if new users opt to share their data. There may also be data removal requests from existing users under the *right to be forgotten* (Shastri et al., 2019). Here, machine unlearning (Cao and Yang, 2015; Bourtoule et al., 2021a) is often the proposed solution by enabling post-hoc removal of data (e.g., PII) from neural models after training.

LLMs are known to memorize and regurgitate personal information and PII (Carlini et al., 2021; Nasr et al., 2023), which is a concrete privacy harm we study. In this literature, little focus has been given to how this may arise dynamically as a part of a machine learning system. In this work, we study how various actions (continually training on more data, re-training with new data, or re-training

[*]Equal senior authorship.

[1]See https://platform.openai.com/docs/guides/fine-tuning/when-to-use-fine-tuning or https://ai.google.dev/gemini-api/docs/model-tuning

[2]We adopt the definition of "memorization" as used at www.genlaw.org/glossary.html

[3]We focus on PII as a more concrete privacy risk, though note that our results likely also extend to broader types of sensitive information. We thus use these terms interchangeably.

Findings of the Association for Computational Linguistics: ACL 2025, pages 18703–18726

after removing data) may influence PII memorization and extraction. We systematically study these operations to determine which improve or worsen the memorization of PII. In particular, we have four **main contributions**[4]:

1. We observe the phenomenon of *assisted memorization*: PII may not be memorized immediately after it is seen, but may be memorized later in training (§5 and Figure 1). We find this is largely influenced by n-gram statistics.
2. We propose a taxonomy of types of PII memorization that arise while training an LLM and show how they manifest (§ 4 and Figure 2).
3. We observe that introducing new PII into training data may worsen extraction of PII (§6.1).
4. We observe that reducing the PII memorization risks for one individual can worsen these risks for another individual (§6.2).

2 Related Work

Membership Inference is one of the most common privacy attacks on neural models (Shokri et al., 2017). Though successful on computer vision models (Yeom et al., 2018; Salem et al., 2018; Sablayrolles et al., 2019; Choquette-Choo et al., 2021; Carlini et al., 2022a; Jagielski et al., 2024), these attacks are not often successful on LLMs (Duan et al., 2024a) which we study. Thus, and because verbatim extraction poses a stronger privacy risk, we focus on *memorization and extraction*.

Memorization & Extraction studies when a text is trained on and generated by a model. This is widely studied (Carlini et al., 2019, 2021, 2022b; Lee et al., 2022; Zhang et al., 2023; Ippolito et al., 2023; Biderman et al., 2023a,b; Kudugunta et al., 2024; Nasr et al., 2023; Borkar, 2023; Chang et al., 2023; Ozdayi et al., 2023; Schwarzschild et al., 2024; Duarte et al., 2024; Wang et al., 2024). These works are often focused on the broad phenomenon, and not the nature of the data, e.g., if it were sensitive as in our work. Relatively fewer works have considered this setting. Huang et al. (2022) study if information about specific entites can be extracted; Panda et al. (2024) study if LLM's can be poisoned to memorize specific PII; Lukas et al. (2023) formalize PII extraction, proposing several attacks and studying the efficacy of various existing defenses; Mireshghallah et al. (2022) and Zeng et al. (2024) study memorization during fine-tuning; and Lehman et al. (2021) found that extracting sensitive data, using simple techniques, from BERT trained on clinical notes was largely unsuccessful. This line of work has become important for practical privacy and memorization audits (Anil et al., 2023; Gemini Team et al., 2023; Dubey, 2024), which also often include PII memorization evaluations (Gemini Team et al., 2023, 2024; Gemma Team et al., 2024a,b; CodeGemma Team et al., 2024).

Dynamics of Memorization. Most related to our work are those exploring memorization throughout training. It is known that language models memorize more as training progresses (Tirumala et al., 2022; Prashanth et al., 2024; Huang et al., 2024) and exhibit forgetting of memorized examples (Jagielski et al., 2022). Biderman et al. (2023a) found that there is not high correlation between memorized sequences within checkpoints of a training run. Duan et al. (2024b) show a similar notion of "latent memorization" but that instead uses Gaussian noise to uncover these latent memories; instead, our "assisted memorization" shows this can happen in normal training runs through only naturally occurring text sequences. The literature so far lacks a clear understanding of the complete memorization landscape throughout training. In our work, we provide a complete taxonomy and uncover novel forms of memorization within training dynamics.

Unlearning Machine unlearning methods have been proposed as an efficient way to erase data from neural networks (Bourtoule et al., 2021b; Izzo et al., 2021; Thudi et al., 2022). These methods are motivated by scenarios where users may request for their data to be removed from a trained model (possibly due to legislative considerations like GDPR (Fabbrini and Celeste, 2020)). While many techniques have been proposed for machine unlearning, we focus on the simple strategy of retraining without relevant data points which is the current gold standard, though it may not be applicable to all practical scenarios (Cooper et al., 2024). Most related to our work are works that show unlearning can cause additional privacy risks: Chen et al. (2021) show this can lead to stronger membership inference attacks and Carlini et al. (2022c); Hayes et al. (2024a) show that unlearning can in-

[4]Code available at https://github.com/jaydeepborkar/Assisted-Memorization

Memorization Category	Extracted at *i-1*	Extracted at *i*
Immediate	✓	N/A
Forgotten	✓	✗
Retained	✓	✓
Assisted	✗	✓

Figure 2: **Taxonomy of memorization for a continuous training setup.** We define immediate, retained, forgotten, and assisted (described in Section 4.1). Note that text classified as assisted memorization may also be forgotten or retained for steps $i + 1$ onwards.

crease membership inference accuracy on other training samples.

3 Experimental Setup

Our goal is to study how memorization of PII manifests during training.[5] This includes continual training or fine-tuning setups in §4 and re-training or unlearning setups in §6. First, we describe our general experimental setup.

Training Setup We use the GPT-2-XL model (Radford et al., 2019), which has 1.5B parameters for our primary experiments, and also experiment with Llama 3 8B (et al., 2024)[6] and Gemma 2B (Gemma Team et al., 2024a). We fine-tune these models with a linear schedule: initial and end learning rate of zero, 500 step warmup, cooldown, and peak learning rate of 2×10^{-5}. We use 1×10^{-2} weight decay and a batch size of 8. We run experiments 5 times, sampling fresh randomness (model weights, data order, etc.) each time.

We fine-tune these models on two datasets. First, we use a modified version of the WikiText-2 dataset (Merity et al., 2016a) to include unique emails from the Enron dataset[7]. We take the entire WikiText-2 dataset and insert E unique email addresses (herein, emails) randomly into each passage. We perform random insertions to eliminate any contextual dependency between the emails and the surrounding text, which the model could otherwise use to predict the emails more accurately. This allows us to study memorization in a worst-case scenario where no contextual cues are available (more details in Appendix A). We concatenate all passages during training and divide them into blocks of 128 tokens. Second, we use the Pile of Law dataset (Henderson et al., 2022) (Appendix C). We ensure no emails were already memorized by querying the base models with the same prompts. Lee et al. (2022) and Kandpal et al. (2022) found data duplication strongly increases memorization. In our study, all emails occur in the training corpus exactly once.

Sampling We closely follow the methodology of Carlini et al. (2021); Nasr et al. (2023). We focus on "extractable memorization" and use ten-token sequences sampled uniformly at random from Common Crawl. We randomly sample a unique set of $25,000$ different prompts for each experiment. We obtain a 256 token output from the model for each prompt and evaluate it for successful extraction. Our method may lead to false negatives; however, this would only underestimate the PII regurgitation, and, we further believe our diverse and large prompt dataset reasonably captures the regurgitation rates. To further minimize false-negatives, where denoted we also evaluate "discoverable" memorization, where we prompt with the exact prefix the model was trained on (Appendix B Section B.2). We use greedy decoding, or top-$k = 40$ sampling when specified.

Defining Memorization and Extraction We primarily use the definition of *extractable memorization* (and, where denoted, *discoverable memorization* (Section B.2)) from Nasr et al. (2023). Herein, we will refer to a success as an extraction, which is whenever an *email* is contained both in the training dataset and a language model's generation. Formally, let $\mathcal{D}$ be the training dataset for a language model M. Let f be a chosen sampling scheme that takes an input text prompt p and returns the conditional generation $s = f_M(p)$. An email e^i is said to be extracted if $e^i \in \mathcal{D}$ and $\exists p : e^i \in f_M(p)$.

Checking for Memorized PII We use a regular expression to identify any emails within the generations that belong to the model's training data. Unlike previous approaches that create a pool of

[5] We do not state or imply [here] that a model "contains" its training data in the sense that there is a copy of that data in the model. Rather, a model memorizes attributes of its training data such that in certain cases it is statistically able to generate such training data when following rules and using information about features of its training data that it does contain.

[6] Llama experiments in this paper were conducted only by parties outside of Google.

[7] https://www.cs.cmu.edu/enron/

generations by filtering based on factors like perplexity and entropy (Carlini et al., 2021), we evaluate all 25,000 generations for memorization.

4 A Dynamic Lens on PII Memorization

Production language models today consist of many training stages (pre-training, post-training, product-specific fine-tuning, etc.) and may be continually updated or refreshed with new data, e.g., to incorporate new human data using RLHF (Stiennon et al., 2020). These stages may incorporate varying degrees of personal information. This raises the question: *how does memorization of sensitive data like PII evolve in this dynamical system?*

Continuous Training Setup. To study this question, we use the simplest setup that generally captures all of the above scenarios: we study memorization throughout supervised fine-tuning. We train a model by keeping the rate of emails seen constant and save checkpoints at regular intervals (for efficiency, only every 10% of training). Details on the dataset construction are in §3.

4.1 Categorizing Memorization Phenomena

Memorization analysis is typically based on *only* the final model, in both academia (Carlini et al., 2022b) and industry (Gemini Team et al., 2024; et al., 2024; Gemma Team et al., 2024b). We now present our taxonomy for dynamic memorization analysis and use it to analyze how memorization manifests throughout continual training.

We begin by looking at the first step of training. There are but two options for any PII seen in this step: for the model to memorize it, or not. We call this type of memorization immediate, since by construction our dataset contains this email exactly once. Now, say this model were trained for another step. This new model may observe new (immediate) memorization. Beyond this, we would expect that the rest of the memorization overlaps with the prior model, which we call retained memorization, similar to analysis in Biderman et al. (2023a). Finally, Jagielski et al. (2022) would tell us that we may also expect some sequences to be forgotten. *However, we observe an additional phenomenon: assisted memorization.* This occurs when PII not memorized at the immediate checkpoint becomes extractable later in training. We discuss this in more detail in § 5. Figure 2 shows our complete memorization taxonomy.

4.2 Experimental Results

Using this taxonomy of immediate, retained, and forgotten memorization (and assisted memorization), we characterize all the extracted emails we observe throughout training (using the setup described above). Our results are shown in Figure 3. We observe that there is a trend that more immediate memorization occurs near the beginning of training, whereas there is a lower rate of immediate memorization later in training. This trend is particularly true for larger models, likely because these models memorize faster.

We also find that models are constantly forgetting. Throughout the entirety of training (including the beginning and end), many models (see Appendix C for more results on other models and datasets) exhibit a cycle of forgetting and immediate memorization. This result sheds new light on the dynamic view of memorization: which samples are memorized by a model may be more a function of stochasticity than previously thought. The choice of which model to release may play a larger role in determining which samples are memorized, due to which samples were forgotten or re-memorized than previously thought due to the stochasticity in data sampling.

Not all memorization occurs immediately. When using our taxonomy to analyze memorizing, we observe that a significant fraction of memorization samples are not classified by these three categories. This leads to another interesting finding: a lot of memorization is *not* immediately memorized. In other words, at a given step, other text that *was not trained on at this step* is now extractable by the model.

Forgetting and Re-Extraction of PII. Our results in Figure 3 show that LLMs do forget some of the previously memorized PII as training progresses. Prior work has shown that some examples memorized early in training may be forgotten after additional training (Jagielski et al., 2022). Further, we also observe that some *forgotten* emails get *re-extracted* when there is n-gram overlap between tokens from the email and tokens in the data during further training. This phenomenon is illustrated in Figure 4, which shows how previously extracted samples that the model later *forgets* can reappear at subsequent checkpoints. Each cell indicates the percentage of emails extracted by both the corresponding checkpoint and the reference checkpoint

(diagonal cell). Since each diagonal cell serves as its own reference, its value is always 1.

5 Assisted Memorization: Training on One's PII Can Reveal Another's

In Figure 3, we see that a large fraction of memorization is assisted. This is especially true later in training, where we observe that more memorization is assisted than immediate, specifically a mean rate of 0.03 for assisted compared to 0.01 for immediate. This finding is not model- or data-specific, as our results in Appendix C (e.g., Figures 17 and 18) show similar trends.

The existence of assisted memorization brings to light a deeper privacy concern. One may expect that data seen earlier is less vulnerable to privacy risks through a form of "recency bias" (implied by forgetting effects). Our findings of assisted memorization, however, show that this may not always be the case; the existence of this effect with sensitive data like PII is of particular concern because it shows that downstream training stages must be careful how they may elicit the extraction of earlier training data. The most common practical scenario for this is in the pre-training/fine-tuning setup that current LLMs undergo. Our results show that fine-tuning even on natural (non-adversarially) constructed training datasets can uncover the extraction of PII in pre-training data. Prior work (Nasr et al., 2023) only showed that this may be possible with adversarial constructions. Pragmatically, our results also show that privacy and memorization audits, especially when PII is of concern, should encompass all data in the training history, and not just data from the most recent training stage.

5.1 Assisted Memorization Is Not Simply Delayed

Above, we found that extraction can be elicited at training steps later than where a piece of sensitive text was seen during training, in what we call assisted memorization. Here, we explore to what degree this assisted memorization is assisted by particular text in the training data, or if it was inevitable and simply delayed.

We find emails that were identified as assisted memorization at various points in training. Our aim is to re-perform training between when they were first seen and when they were later extractable by selecting entirely fresh data from the remainder of the (unseen) training dataset. Then, we can observe if only this unique set of data elicited the memorization or if any batch could.

We know when data samples were first seen from data sampling. Then, we must identify exactly when each email became extractable, as any training beyond this may lead to forgetting. Given that we only checkpoint our models every 10% of training, for efficiency, we do not have this a priori. To determine this, we use a binary search, performing an extraction test on each iteration of the search. This significantly reduces the overhead as the extraction test is expensive (recall we prompt the model thousands of times as described in §3).

Overall, we run this procedure on four unique emails and with seven trials each. We find that emails became extractable in only $35.7\% \pm 15.9$ of them on average. While this refutes the idea that there may be a single unique set of data that leads to assisted memorization, this shows that most sets of data do not lead to it. Next, we explore what characteristics the successful trials share.

5.2 Assisted Memorization Is Triggered by Training on Specific n-grams

Our analysis here is inspired by Lee et al. (2022), who show that data repetitions (duplication) heavily influence memorization of text. While our data setup in §3 has no exact duplicates of these emails, there can still be overlaps of important n-grams.

Causally Removing n-grams. To study this, we perform a causal intervention whereby we remove all training sequences that have high n-gram overlap with emails identified as assisted memorization. We use a similar setup to the previous §5.1 except that we notably remove any text that overlaps with the assisted memorized emails. For each trial of this experiment, we select a different checkpoint M_i throughout our continuous fine-tuning setup; let $\mathcal{D}_i$ be the set of training sequences used to train M_i from M_{i-1}. We take all emails identified as assisted memorization on M_i; for each, we construct a simple regex-based filter that checks for names in the email address based on common email formatting patterns (e.g., name@gmail.com or firstname.lastname@gmail.com). We use these regex filters to remove any text in $\mathcal{D}_i$ and then retrain M_i from M_{i-1} on this new dataset.

Across all 30 checkpoints and 5 seeds, we find a total of 177 emails that were assisted memorized. After intervening to remove overlapping n-grams from batch D_i, all but 10 of these assisted memo-

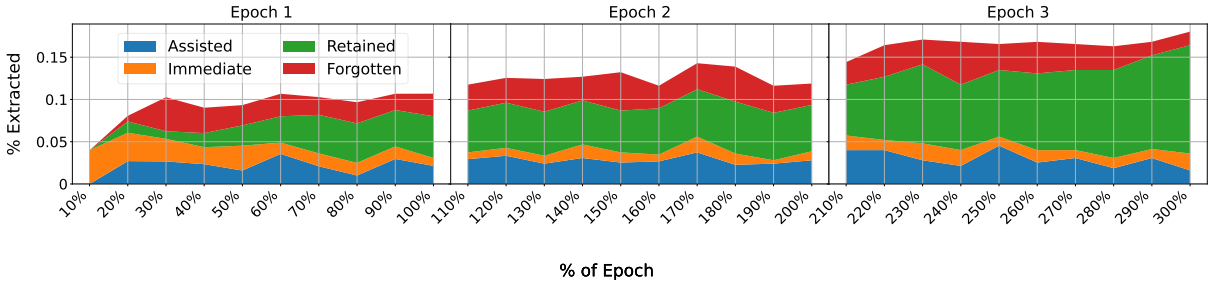

Figure 3: **Tracking memorization throughout training with our taxonomy.*** The stacked bars show how many newly memorized emails are immediate, retained, and assisted, while red denotes forgotten emails since the last checkpoint. We see large amounts of assisted memorization occurring later in training, underscoring that PII is not always memorized immediately. *Takeaway:* memorization is more dynamic and stochastic than often assumed, with ongoing cycles of forgotten and newly assisted emails. *Total number of extracted emails at each checkpoint are normalized by the number of emails seen until that checkpoint.

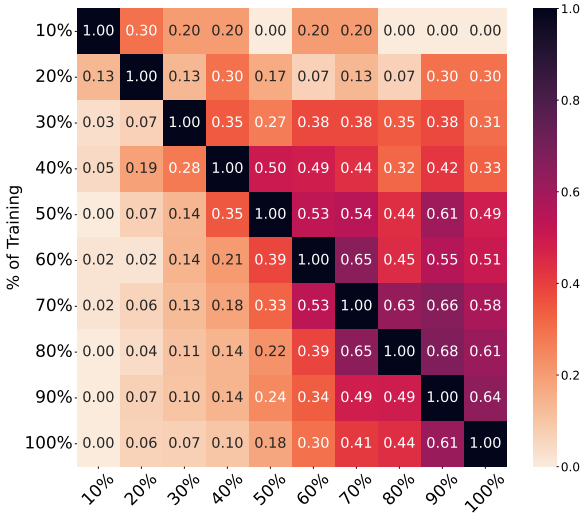

Figure 4: **Forgotten PII is re-extracted later.** The diagonal values d_{ii} (reference cells) represent the total extraction at each checkpoint; off-diagonal cells show the fraction of emails from the reference cell that are also memorized in the current cell. *Takeaway:* memorized PII can sometimes slip out of memory, only to reappear once certain overlapping tokens occur in future training steps.

rized emails were no longer memorized.

Features Associated with Memorization Next, we ask: when multiple emails share a `firstname`, why might a particular email with a different `lastname` get assisted memorized over another? For example, why might `john.mccarthy@gmail.com` be memorized over `john.williams@gmail.com`. We train a simple logistic regression model on features capturing n-grams overlaps, last-name counts, and domain counts for all assisted memorized emails (positives) and those not memorized (negatives). More details are in Appendix D.

Our logistic regression model is trained to predict assisted memorized emails from a dataset consisting of these emails labeled as positive, and other emails sharing the same `firstname` but a different `lastname` as negatives. We use a standard 5-way cross validation setup with 10 trials. Full details are in Appendix D. The model achieves a precision of 0.937 and recall of 0.874 indicating high success.

In Figure 5, we visualized the logistic regression model's score against the email likelihood from M, computed against the successful prompt that led to extraction. This shows that assisted memorization emails tend to be well classified from these simple features. We observe that n-gram statistics were the most important feature, further supporting our conclusions above (see Table 1 of Appendix D where we report the feature weights).

6 Do PII Opt-ins/Opt-outs Impact Extraction?

6.1 Contributing More Data via Opt-ins

If many new users opt-in to contribute data to a model, then the model owner may want to incorporate new information (and sometimes, new PII) into the finetuning pipeline. One of the simplest ways to do this is by adding the new PII to existing training data and re-finetuning the model from scratch. From our results in §5, we know that continuing to train a model on additional PII could lead to increased extractability of previously unextracted PII. In this section, we study how retraining with additional PII changes the extractability of prior data.

Setup To mimic the above scenario, we design a **Retraining Experiment** where we add more

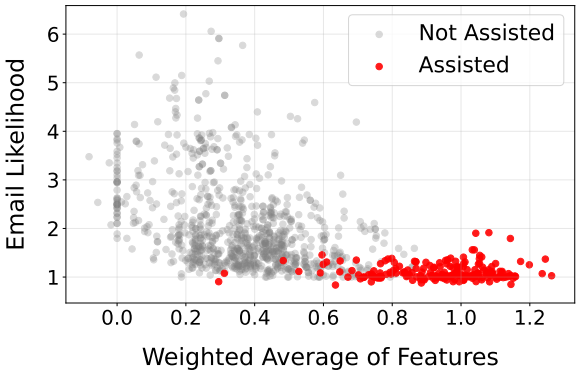

Figure 5: **Overlap features predict which emails are assisted memorized.** We plot a logistic-regression score (x-axis) vs. conditional likelihood (y-axis). Emails that become assisted memorized (red) exhibit higher n-grams overlap (i.e. higher model score), whereas those not memorized (grey) have lower overlap. *Takeaway:* overlapping n-grams in future training data strongly drive which PII is triggered to appear in the model's output.

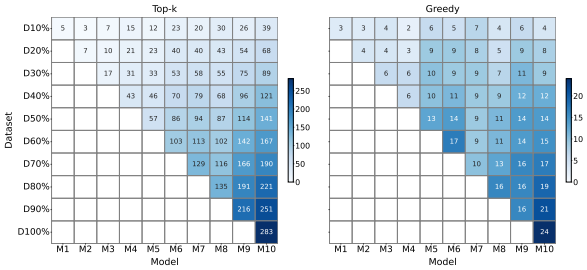

Figure 6: **Adding more PII leads to more extraction.** Each row corresponds to a dataset $D_{x\%}$, and each column corresponds to the model M_j trained with $j \times 10\%$ of the emails. The values show how many emails in $D_{x\%}$ are extracted by M_j. *Takeaway:* introducing new PII during re-finetuning (moving along the x-axis) also increases the extraction *of old PII* that was already present in the training set. This effect can increase extraction by a factor of over 7× in our settings, as seen in the extraction of emails in $D_{10\%}$ from M_{10}.

emails to the existing dataset and re-finetune the model on the updated dataset. We write $D_{x\%}$ as the finetuning dataset containing $x\%$ of the emails from the global set of emails X. We construct 10 different finetuning datasets containing increasing amounts of emails: $D_{10\%}, D_{20\%}, \cdots, D_{100\%}$. In $D_{x\%}$, we include $x\%$ of the global pool of emails X, such that, if $a < b$, all emails that are found in $D_{a\%}$ are also found in $D_{b\%}$. Before constructing these datasets, we randomly shuffle the emails in X to ensure a uniform distribution of emails in each dataset.

Next, we train ten distinct models M_1 to M_{10}, where M_i is trained on $D_{10i\%}$ for three epochs, following the same training setup described in §3. We highlight that the only change between these models is the additional emails. Otherwise, we use the same training process and the same prompts for all models when decoding.

Adding More PII Increases Extraction of Existing PII. We report the results of our experiment in Figure 6, for models finetuned for three epochs (more results in Appendix E). We highlight two major findings.

First, we find that the number of extracted emails increases substantially with the amount of PII contained in the model's fine-tuning set. This can be seen on the diagonals of Figure 6, which show the total amount of PII extracted from the relevant model. For top-k sampling, we see that 283 emails are extracted from M_{10}, compared to only 57 at M_5, which was trained on half as many emails—the increase in extraction from top-k sampling is superlinear in the fraction of emails included in the model's finetuning set. The increase is still substantial, but not superlinear, for greedy sampling.

Our second and main finding is that the inclusion of more PII leads to *existing* PII being at higher risk of extraction from top-k sampling. This can be seen from the general positive trend in extracted emails for each dataset $D_{x\%}$ along the x axis. To validate this result, we run a binomial hypothesis test, for whether top-k sampling extracts more emails from $D_{i\%}$ when run on M_j ($j > i$) than when run on M_i. With 45 such comparisons, 41 show more extraction for models which see more emails ($p < 10^{-8}$, and $p < 10^{-4}$ for 1 and 2 epochs).

6.2 Protecting PII via Opt Outs

As data opt-outs are becoming increasingly common on the web (LinkedIn, 2023), we first study how removing a user's PII from the training data can inadvertently trigger the extraction of additional PII. We then investigate factors that correlate to PII becoming extractable once similar PII is removed.

Setup We study the simplest unlearning technique, often referred to as *exact machine unlearning* (Bourtoule et al., 2021a): removing all relevant PII from the dataset and retraining, or as here re-fine-tuning, the model. This may be triggered if users submit an opt-out request. Since retraining after each request is expensive, model owners may

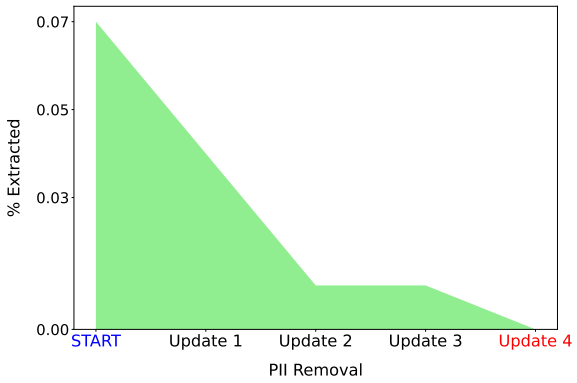

Figure 7: Removing extracted PII from the training data and retraining can lead to new memorized PII. After four removal-and-retrain cycles (Update 1–4), no additional PII is extracted under the same 25k prompts and greedy decoding. START denotes the original model.

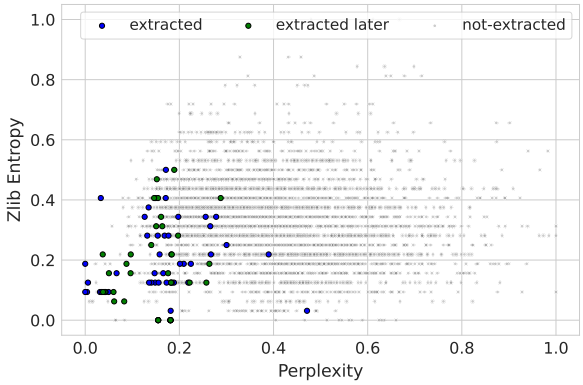

Figure 8: Perplexity and zlib entropy of memorized emails. Emails extracted in the initial model (blue) and emails extracted in later re-finetuned models (green) have lower perplexities than emails that were never extracted by any model (grey). This clustering suggests that the newly-extracted (green) emails were near the threshold of memorization from the outset.

collect and process these requests in batches.

Following a protocol similar to Carlini et al. (2022c), our experimental procedure is: **(1) Extraction:** Prompt the current model $\mathcal{M}$ with 25,000 fixed prompts and sample using greedy decoding to identify memorized emails. Let E be the set of extracted emails. **(2) Removal:** Remove E from D and re-finetune the base model on $D \setminus E$, producing a new model $\hat{\mathcal{M}}$. **(3) Repeat:** Prompt $\hat{\mathcal{M}}$ again with the same prompts, discovering any newly memorized emails $\hat{E}$. We iterate until no more emails are extracted using this fixed set of prompts and decoding strategy.

Protecting One Person's PII May Leak Another's As mentioned above, in each iteration, we (1) prompt the current model $\mathcal{M}$ (trained on dataset D) with 25,000 fixed prompts, (2) remove any newly discovered memorized emails E from D, and (3) re-finetune the base model on $D \setminus E$. Figure 7 illustrates four such rounds (START through Update 4). While the first update successfully removes the previously identified emails from the set of extracted PII, it simultaneously extracts a *new* set of emails. By Update 4, no additional emails are discovered under these prompts and greedy decoding, although changing prompts or sampling strategies could still reveal further memorization. Our results confirm that this *layered memorization*—called the Onion Effect by prior work on image classifiers (Carlini et al., 2022c)—extends to language models: removing one layer of memorized PII exposes a second layer, and so forth.

Removing Random Emails. We next conduct a similar experiment but remove a random subset of emails instead of the ones that are discovered through extraction. Specifically, we sample 10% of the total emails in D uniformly at random and call this set E. We then fine-tune a new model $\hat{\mathcal{M}}$ on $D \setminus E$. Prompting $\hat{\mathcal{M}}$ with the same 25,000 prompts and sampling with greedy decoding yields a new set of extracted emails $\hat{E}$. Thus, *randomly removing* data can similarly expose new PII, underscoring how unlearning updates can inadvertently introduce new privacy risks.

Controlling for Randomness During Training. A natural question is whether any newly extracted emails simply result from any randomness when retraining a new model. For instance, models trained with the same data order, same parameter initialization, and same hyperparameters could still differ during inference as GPU operations are nondeterministic (Jagielski et al., 2020). We want to ensure that new extractions are solely the result of removing particular emails. To this end, we train five such new models and extract emails by feeding the exact same prompts that we give to our original model ($\mathcal{M}$) and the models trained after removing extracted and randomly sampled emails ($\hat{\mathcal{M}}$). We sample all three sets of models with greedy decoding and compare which emails were extracted. Across all five trials and for both types of removals (removing extracted emails and removing them randomly), the models re-finetuned-after-removal reveal strictly more *unique* PII than these fresh counterparts. Hence, the effect is not merely a product of random training fluctuations but rather

an outcome of selectively removing data from D.

PII on the Verge of Memorization Surfaces After Others Are Removed Because we use a fixed set of prompts and greedy decoding, we hypothesize that newly extracted emails in each unlearning round were already *close* to being memorized under the original model. In other words, these emails were initially "hidden" behind a first layer of memorized PII. Once the first layer of emails is removed, these nearly extractable emails become more vulnerable.

To investigate this, we compare the perplexity of the initial model on three categories of emails: *(i)* those extracted in the initial model, *(ii)* those that are extracted in subsequent rounds of removal and refinetuning and *(iii)* those never extracted by any model. We also measure their zlib entropy, a compression-based proxy for memorization (Carlini et al., 2021; Prashanth et al., 2024; loup Gailly and Adler). As shown in Figure 8, newly-extracted emails (green) cluster with those initially extracted (blue), indicating that both groups have lower perplexity compared to never-extracted emails (grey). This supports our hypothesis: once one layer of extracted PII is removed from the training set, the next-likeliest set of emails crosses the threshold into extraction. Iterating this process eventually exhausts these "hidden layers," although more sophisticated prompts or sampling strategies could still uncover additional memorization.

7 Conclusion

We study how the actions of continually training on more data, re-training with new data, or re-training after removing data can have ripple effects for privacy. In particular, we propose the phenomenon of Assisted Memorization where examples that aren't extracted at existing checkpoints can get extracted later. This could create a false impression of privacy for examples that don't get extracted at a particular checkpoint, as training further on similar-appearing examples could lead to their extraction. We also find that including more PII in the training data can degrade privacy of existing PII by putting them at a higher risk of extraction. Furthermore, removing particular PII examples from training data could cause other examples to be extracted. This underscores the need for more holistic audits for memorization, where examples that aren't extracted at a particular timepoint are also evaluated for any potential risks.

Limitations

In this study, we use emails as an example of PII because they are a common form of personal information and can be readily studied using publicly available datasets, e.g., the Enron corpus. We do not examine other forms of PII, such as credit card numbers or mailing addresses, partly because they are not publicly available. However, analyzing these types of PII is important to determine whether certain categories are more vulnerable to the memorization risks identified here. We believe that our methods will generalize to other forms of PII with minor adjustments. We also observe a phenomenon akin to *onion memorization* (Carlini et al., 2022c), where removing particular emails from the dataset and retraining the model (*exact unlearning* (Bourtoule et al., 2021b)) can cause new emails to be extracted. A promising direction is to investigate whether this effect persists under *approximate* unlearning techniques (e.g., (Hayes et al., 2024a)), where the model is not fully retrained from scratch. Furthermore, our focus here is solely on extraction risks for training-data emails, but other generated or partially memorized emails could also pose privacy concerns—particularly if they can serve as keys to uncover additional information about specific individuals.

Ethics Statement

We rely on the publicly available Enron Corpus to create our fine-tuning datasets, acknowledging that some of its contents may include sensitive or personally identifiable information. To mitigate privacy risks, we follow standard diligence practices for data handling. While no additional raw text or private details are disclosed beyond those already publicly released, we analyze memorization specifically to highlight risks inherent in large language models, rather than to reveal more personal data. Our experiments use established public models and datasets (GPT-2 family, Gemma 2B, Llama 3 8B, Wikitext, and Pile of Law) to facilitate reproducibility while maintaining responsible data practices. We align our work with accepted norms for ethical use of legacy datasets like Enron and emphasize the importance of privacy-preserving training and unlearning techniques for future systems.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. 2023a. Emergent and predictable memorization in large language models. *Preprint*, arXiv:2304.11158.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023b. Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Jaydeep Borkar. 2023. What can we learn from data leakage and unlearning for law? *Preprint*, arXiv:2307.10476.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021a. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021b. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022a. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022b. Quantifying memorization across neural language models. *arXiv preprint*.

Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. 2022c. The privacy onion effect: Memorization is relative. In *Advances in Neural Information Processing Systems*, volume 35, pages 13263–13276. Curran Associates, Inc.

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, page 267–284, USA. USENIX Association.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar 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. USENIX Association.

Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to ChatGPT/GPT-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327, Singapore. Association for Computational Linguistics.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2021. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 896–911, New York, NY, USA. Association for Computing Machinery.

Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR.

CodeGemma Team, Heri Zhao, Jeffrey Hui, Joshua Howland, Nam Nguyen, Siqi Zuo, Andrea Hu, Christopher A Choquette-Choo, Jingyue Shen, Joe Kelley, et al. 2024. Codegemma: Open code models based on gemma. *arXiv preprint arXiv:2406.11409*.

A. Feder Cooper, Christopher A. Choquette-Choo, Miranda Bogen, Matthew Jagielski, Katja Filippova, Ken Ziyu Liu, Alexandra Chouldechova, Jamie Hayes, Yangsibo Huang, Niloofar Mireshghallah, Ilia Shumailov, Eleni Triantafillou, Peter Kairouz, Nicole Mitchell, Percy Liang, Daniel E. Ho, Yejin Choi, Sanmi Koyejo, Fernando Delgado, James Grimmelmann, Vitaly Shmatikov, Christopher De Sa, Solon Barocas, Amy Cyphert, Mark Lemley, danah boyd, Jennifer Wortman Vaughan, Miles Brundage, David Bau, Seth Neel, Abigail Z. Jacobs, Andreas Terzis, Hanna Wallach, Nicolas Papernot, and Katherine Lee. 2024. Machine unlearning doesn't do what you think: Lessons for generative ai policy, research, and practice. *Preprint*, arXiv:2412.06966.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh

Hajishirzi. 2024a. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

Sunny Duan, Mikail Khona, Abhiram Iyer, Rylan Schaeffer, and Ila R Fiete. 2024b. Uncovering latent memories: Assessing data leakage and memorization patterns in frontier ai models. *Preprint*, arXiv:2406.14549.

André V. Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. 2024. De-cop: detecting copyrighted content in language models training data. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Kush Dubey. 2024. Evaluating the fairness of task-adaptive pretraining on unlabeled test data before few-shot text classification. In *Proceedings of the 2nd GenBench Workshop on Generalisation (Benchmarking) in NLP*, pages 1–26, Miami, Florida, USA. Association for Computational Linguistics.

Abhimanyu Dubey et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Federico Fabbrini and Edoardo Celeste. 2020. The right to be forgotten in the digital age: The challenges of data protection beyond borders. *German Law Journal*, 21(S1):55–65.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

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

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024a. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024b. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. 2024a. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. *arXiv preprint arXiv:2403.01218*.

Jamie Hayes, Marika Swanberg, Harsh Chaudhari, Itay Yona, and Ilia Shumailov. 2024b. Measuring memorization through probabilistic discoverable extraction. *Preprint*, arXiv:2410.19482.

Peter Henderson, Mark Simon Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel E. Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256GB open-source legal dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jing Huang, Diyi Yang, and Christopher Potts. 2024. Demystifying verbatim memorization in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10711–10732, Miami, Florida, USA. Association for Computational Linguistics.

Daphne Ippolito, Florian Tramer, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher Choquette Choo, and Nicholas Carlini. 2023. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53, Prague, Czechia. Association for Computational Linguistics.

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. 2021. Approximate data deletion from machine learning models. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2008–2016. PMLR.

Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. High accuracy and high fidelity extraction of neural networks. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA. USENIX Association.

Matthew Jagielski, Milad Nasr, Katherine Lee, Christopher A Choquette-Choo, Nicholas Carlini, and Florian Tramer. 2024. Students parrot their teachers:

Membership inference on model distillation. *Advances in Neural Information Processing Systems*, 36.

Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, et al. 2022. Measuring forgetting of memorized training examples. *arXiv preprint arXiv:2207.00099*.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun KIM, Stanley Jungkyu Choi, and Minjoon Seo. 2022. Towards continual knowledge learning of language models. In *International Conference on Learning Representations*.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4764–4780, Seattle, United States. Association for Computational Linguistics.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10697–10707. PMLR.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pretraining of language models. In *The Eleventh International Conference on Learning Representations*.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2024. Madlad-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems*, 36.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron Wallace. 2021. Does BERT pretrained on clinical notes reveal sensitive data? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 946–959, Online. Association for Computational Linguistics.

LinkedIn. 2023. Linkedin's data opt-out information. [Online; accessed 14-Feb-2025].

Jean loup Gailly and Mark Adler. zlib compression library.

N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Beguelin. 2023. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363, Los Alamitos, CA, USA. IEEE Computer Society.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016a. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016b. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. 2022. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1816–1826, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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. *Preprint*, arXiv:2311.17035.

Mustafa Ozdayi, Charith Peris, Jack FitzGerald, Christophe Dupuy, Jimit Majmudar, Haidar Khan, Rahil Parikh, and Rahul Gupta. 2023. Controlling the extraction of memorized data from large language models via prompt-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1512–1521, Toronto, Canada. Association for Computational Linguistics.

Ashwinee Panda, Christopher A. Choquette-Choo, Zhengming Zhang, Yaoqing Yang, and Prateek Mittal. 2024. Teach llms to phish: Stealing private information from language models. *Preprint*, arXiv:2403.00871.

USVSN Sai Prashanth, Alvin Deng, Kyle O'Brien, Jyothir S V au2, Mohammad Aflah Khan, Jaydeep Borkar, Christopher A. Choquette-Choo, Jacob Ray Fuehne, Stella Biderman, Tracy Ke, Katherine Lee, and Naomi Saphra. 2024. Recite, reconstruct, recollect: Memorization in lms as a multifaceted phenomenon. *Preprint*, arXiv:2406.17746.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for

language models. In *The Eleventh International Conference on Learning Representations*.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR.

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*.

Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary Lipton, and J. Zico Kolter. 2024. Rethinking llm memorization through the lens of adversarial compression. In *Advances in Neural Information Processing Systems*, volume 37, pages 56244–56267. Curran Associates, Inc.

Supreeth Shastri, Melissa Wasserman, and Vijay Chidambaram. 2019. The seven sins of Personal-Data processing systems under GDPR. In *11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19)*, Renton, WA. USENIX Association.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. 2022. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4007–4022, Boston, MA. USENIX Association.

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

Thomas Vakili, Anastasios Lamproudis, Aron Henriksson, and Hercules Dalianis. 2022. Downstream task performance of BERT models pre-trained using automatically de-identified clinical data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4245–4252, Marseille, France. European Language Resources Association.

Zhepeng Wang, Runxue Bao, Yawen Wu, Jackson Taylor, Cao Xiao, Feng Zheng, Weiwen Jiang, Shangqian Gao, and Yanfu Zhang. 2024. Unlocking memorization in large language models with dynamic soft prompting. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9782–9796, Miami, Florida, USA. Association for Computational Linguistics.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE.

Shenglai Zeng, Yaxin Li, Jie Ren, Yiding Liu, Han Xu, Pengfei He, Yue Xing, Shuaiqiang Wang, Jiliang Tang, and Dawei Yin. 2024. Exploring memorization in fine-tuned language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3917–3948, Bangkok, Thailand. Association for Computational Linguistics.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramer, and Nicholas Carlini. 2023. Counterfactual memorization in neural language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 39321–39362. Curran Associates, Inc.

A More Details on Dataset Construction

While we insert emails into each message at random positions to study the worst-case scenario for memorization, we also want to make sure that the utility of our fine-tuned model is not degraded. To this end, we compare the perplexity values of the original and fine-tuned models on a held-out WikiText-2, as well as a new WikiText-103 (Merity et al., 2016b) test dataset. We compute perplexity values using a sliding window of 1024 tokens (context window of GPT-2 XL). The perplexity of the base GPT-2 XL model on the WikiText-2 test set was 15.20, while that of the fine-tuned model was 11.35. The perplexity of the base model on the WikiText-103 set was 16.49, and the fine-tuned model had a perplexity value of 13.23. These values indicate that the utility of our model is not degraded post-fine-tuning.

B Hyperparameters that Influence PII Extraction

B.1 Greedy vs. Top-*k* Sampling

Model owners can employ either deterministic decoding such as greedy or stochastic sampling methods (such as top-*k* (Fan et al., 2018) or top-*p* (Holtzman et al., 2020)) to improve the quality of the generated text. Several commercial APIs providing text-generation access to models such as ChatGPT[8], Gemini[9], and Claude[10] use a combination of `top-k` and `top-p` parameters to generate text and extraction rates vary across sampling schemes (Hayes et al., 2024b). This makes it essential to study how PII extraction varies across different sampling methods. We find that we can extract significantly more PII using top-*k* sampling than greedy decoding.

We draw the following comparisons: (1) The ratio of total emails extracted using top-*k* sampling compared to greedy decoding; (2) Total emails extracted using a fixed set of 25,000 prompts for both sampling methods; and (3) Total emails generated by both sampling methods when conditioned on same 25,000 prompts.

It can be seen in Figure 9 that top-*k* can extract emails over 800 times higher than greedy decoding. Top-*k* also consistently generates more unique emails than greedy. Model owners might employ top-*k* sampling as it produces more diverse and higher-quality text compared to greedy. However, this approach may pose privacy risks, such as increased memorization and leakage of personal information.

B.2 Prompting

We fine-tune our model on full WikiText-2 with Enron emails in it for 20 epochs and prompt after every epoch in the following manner: (1) **Extractable Prompting** using random ten-token prompts sampled from Common Crawl (as mentioned in § 3), and (2) **Discoverable Prompting** where we prompt with prefixes that occur before an email in the training data (using the definition of *discoverable memorization* from Nasr et al. (2023))

As observed in Figure 10, we find that for both greedy decoding and top-*k* sampling, extractable memorization is more than discoverable memorization in the initial epochs. However, discoverable memorization starts increasing significantly after epoch 5 for greedy decoding and epoch 7 for top-*k* sampling. By the end of the 20th epoch, discoverable memorization is over 92% more than extractable memorization.

C More Results on PII Memorization in Continuous Training.

More results from § 4: We fine-tune various models on two datasets—Wikitext and the Pile of Law—and show that our findings are generalizable. We only use greedy decoding for sampling from these models.

GPT-2 XL trained on the Pile of Law dataset: Figure 11 shows that our results are generalizable also on the Pile of Law dataset (Henderson et al., 2022). We extract the `congressional_hearings` instance from the dataset and insert enron emails in it according to our setup in § 3 while keeping the total number of tokens in the dataset the same as our original Wikitext dataset.

Llama3 8B and Gemma 2B models trained on our original dataset (WikiText + Enron emails): Our results generalize to the current state-of-the-art models, including Llama3 with 8B parameters (Figure 12) and Gemma 2B base model (Gemma Team et al., 2024a) (Figure 13).

GPT-2 Large 774M, Medium 355M, and Small 124M models trained on our original dataset

[8] https://platform.openai.com/docs/guides/text-generation

[9] https://ai.google.dev/gemini-api/docs/text-generation?lang=python

[10] https://docs.anthropic.com/en/api/complete

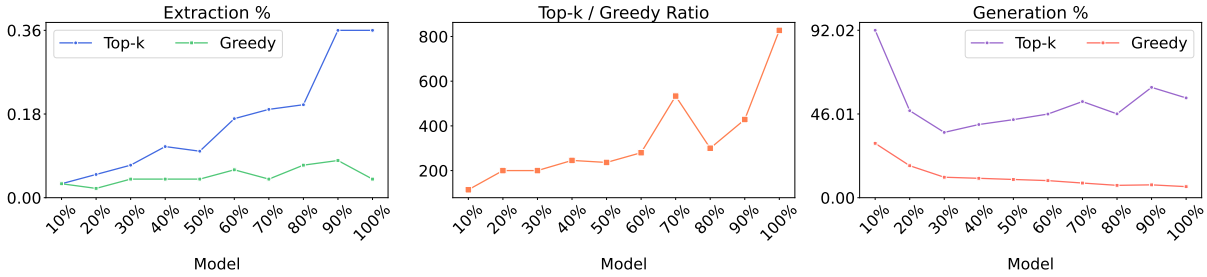

Figure 9: (Left) We can extract significantly more emails with top-k than with greedy decoding using the same set of prompts. (Middle) We can extract up to 800 times more emails using top-k. (Right) top-k generates more emails than greedy for the same amount of emails seen during training. The x-axis denotes a separate model obtained after adding an additional 10% of total emails in the training data.

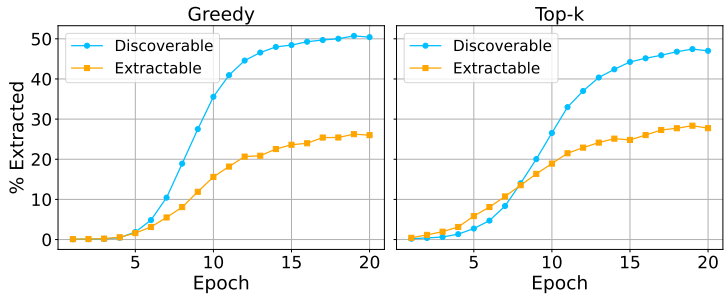

Figure 10: Comparing extractable memorization with discoverable memorization over 20 epochs.

(WikiText + Enron emails): We also train the remaining members from the GPT-2 model family: Large (Figure 14), Medium (Figure 15), and Small (Figure 16). We observe that assisted memorization becomes less prominent in smaller models.

Rate of immediate vs. assisted memorization: We find that the rate of assisted memorization is higher than that of immediate memorization and the difference increases as training progresses. Figure 17 & Figure 18 show this trend for different models.

Forgetting of immediate vs. assisted memorized examples: We do not observe any significant difference between the forgetting rates of both. Figure 19 & Figure 20 show this for GPT-2 XL, Figure 21 shows this for Gemma 2B, and Figure 22 shows this for Llama3 8B.

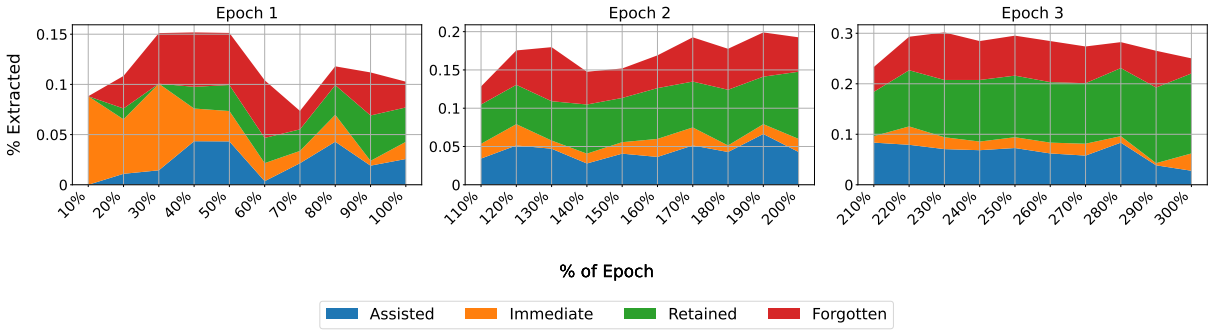

Figure 11: Different memorization categories during continuous training for GPT-2 XL trained on the Pile of Law + Enron emails.

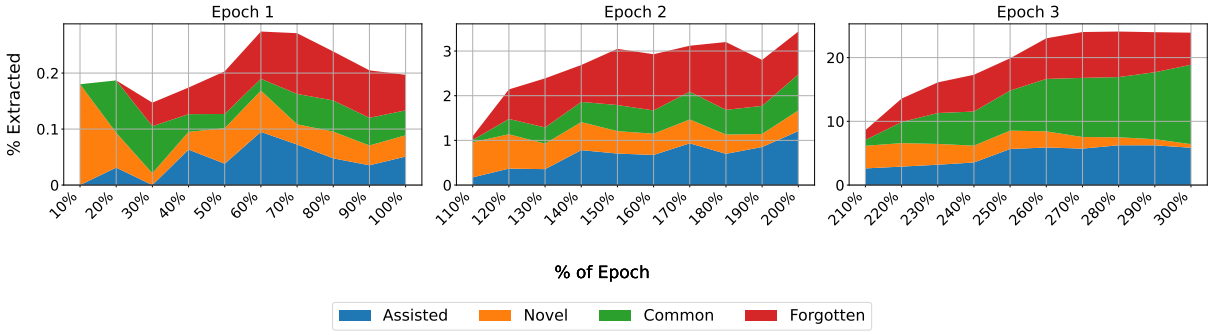

Figure 12: Different memorization categories during continuous training for Llama3 8B trained on WikiText + Enron emails.

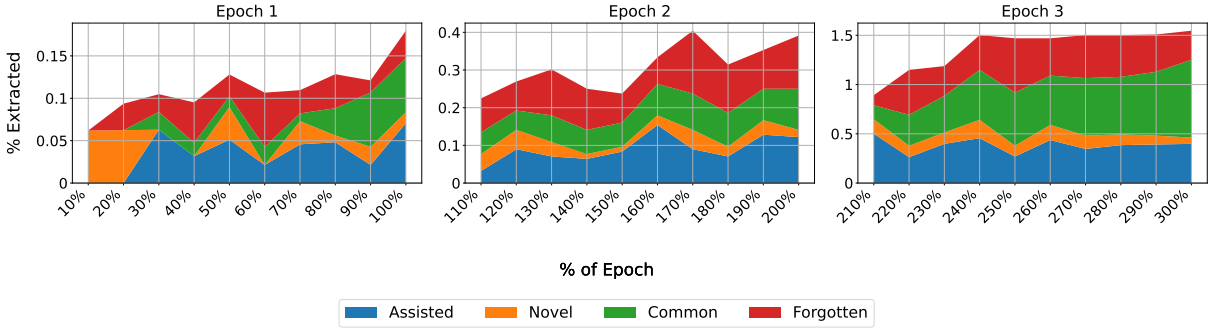

Figure 13: Different memorization categories during continuous training for Gemma 2B trained on WikiText + Enron emails.

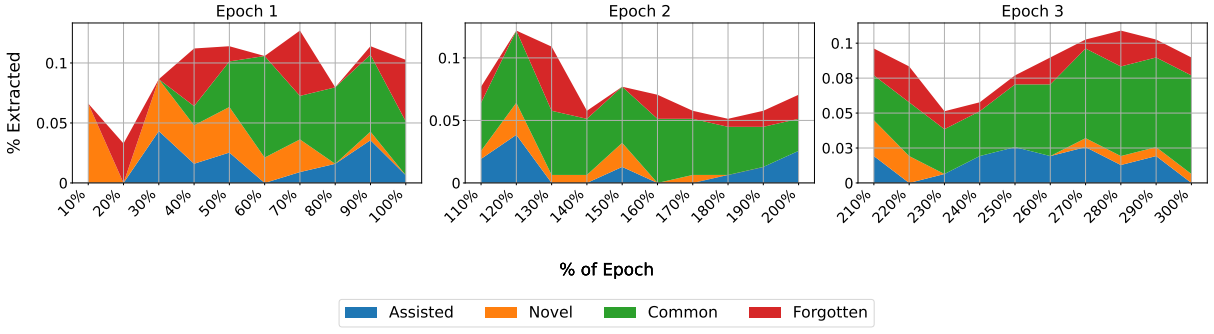

Figure 14: Different memorization categories during continuous training for GPT-2 Large trained on WikiText + Enron emails.

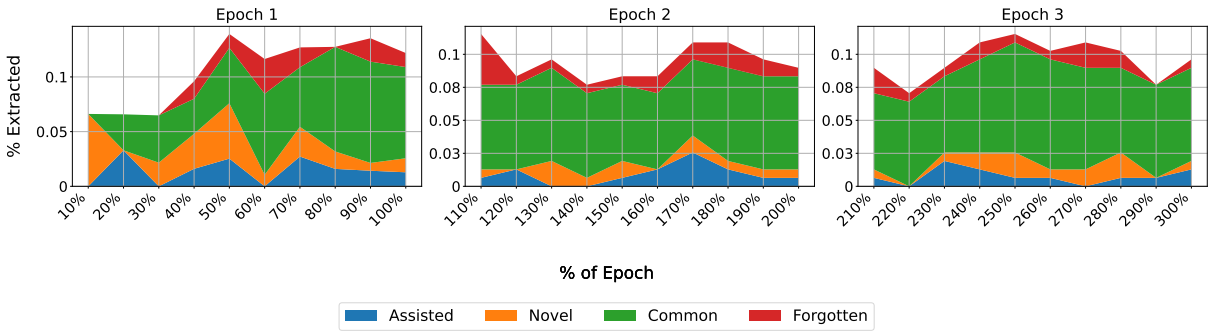

Figure 15: Different memorization categories during continuous training for GPT-2 Medium trained on WikiText + Enron emails.

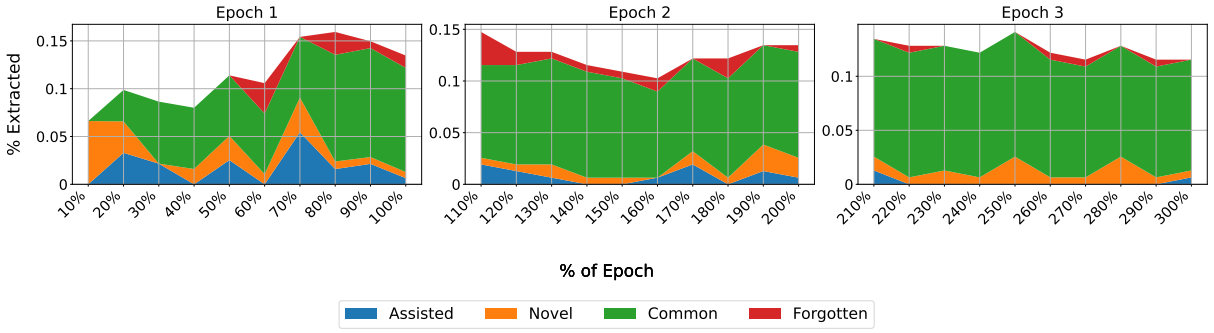

Figure 16: Different memorization categories during continuous training for GPT-2 Small trained on WikiText + Enron emails.

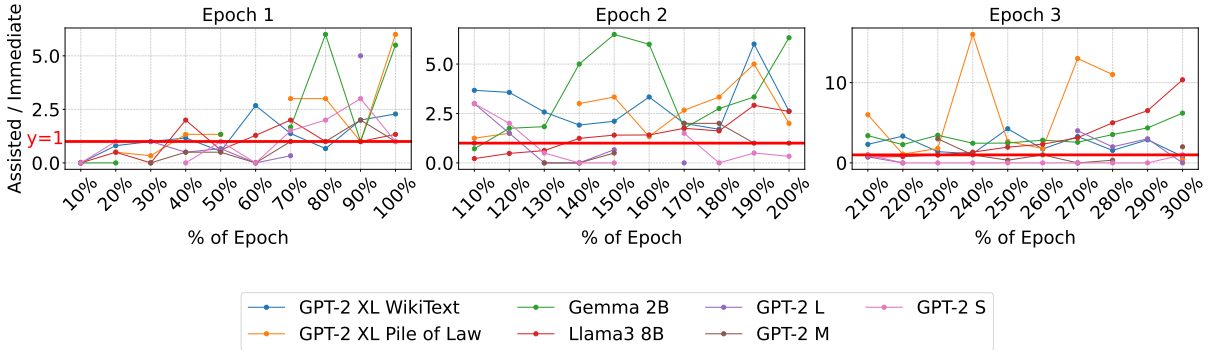

Figure 17: Ratio of the rate of assisted to immediate memorization. We observe that a large fraction of memorization in multiple models is assisted. Note that some models at specific checkpoints had no immediate memorization.

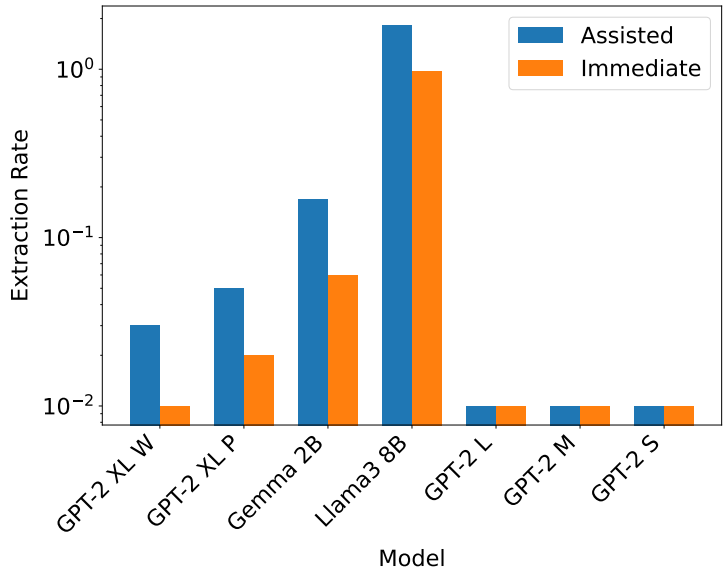

Figure 18: Extraction rates for assisted and immediate memorization denoting the area under curve when continuously trained for three epochs. GPT-2 XL W and P denote the models trained on WikiText and Pile of Law respectively. On average, we observe models have equal or more assisted memorization.

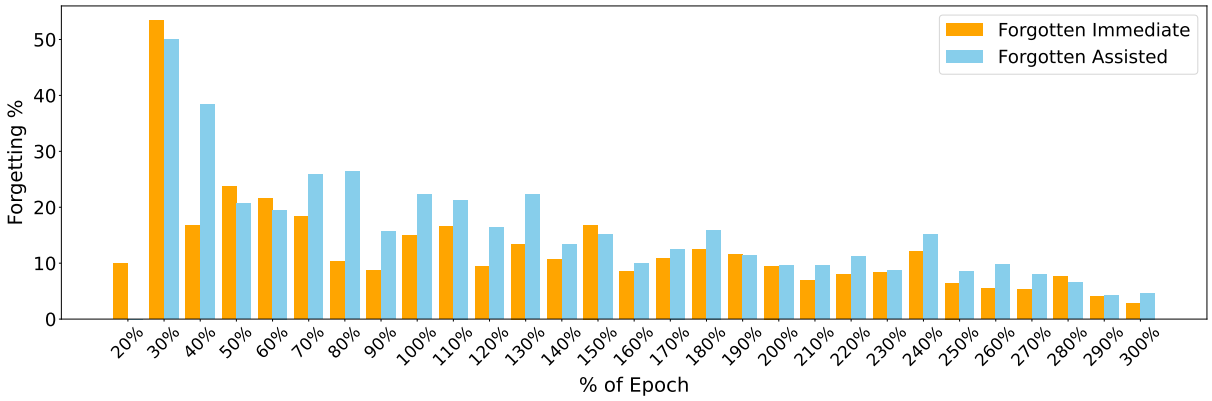

Figure 19: Forgetting rates for GPT-2 XL trained on WikiText + Enron emails. We do not observe any notable difference in the forgetting rates, with assisted (15.54%) being marginally higher than immediate (12.08%).

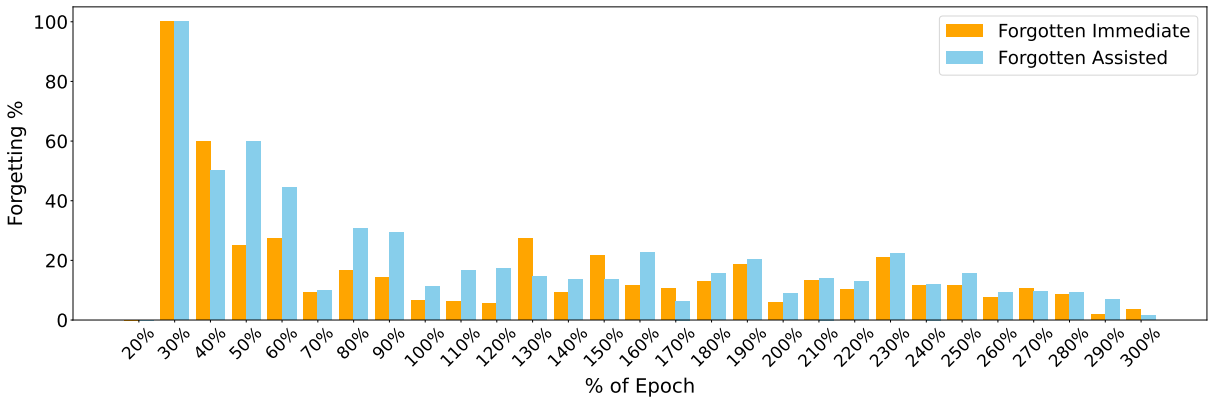

Figure 20: Forgetting rates for GPT-2 XL trained on the Pile of Law + Enron emails. We do not observe any notable difference in the forgetting rates, with assisted (21.06%) being marginally higher than immediate (16.05%).

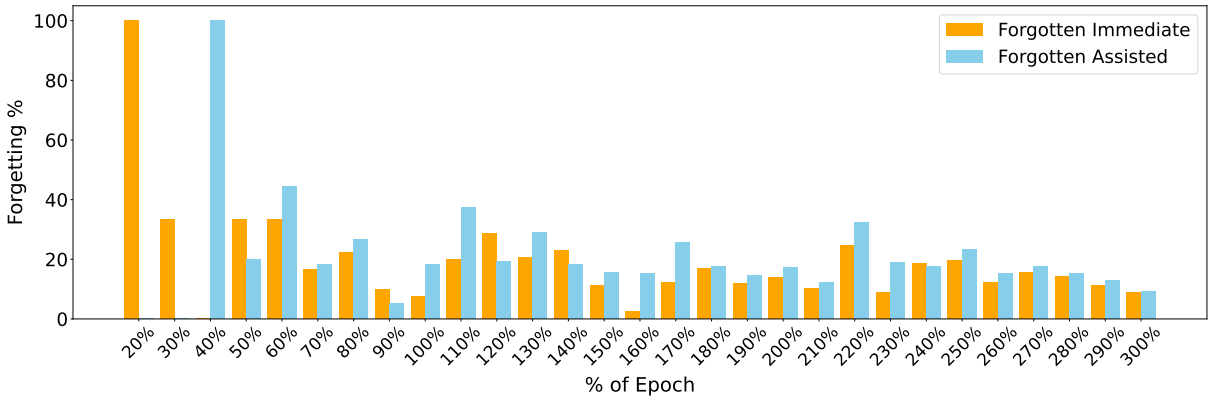

Figure 21: Forgetting rates for Gemma 2B trained on WikiText + Enron emails. We do not observe any notable difference in the forgetting rates, with assisted (20.69%) being marginally higher than immediate (18.05%).

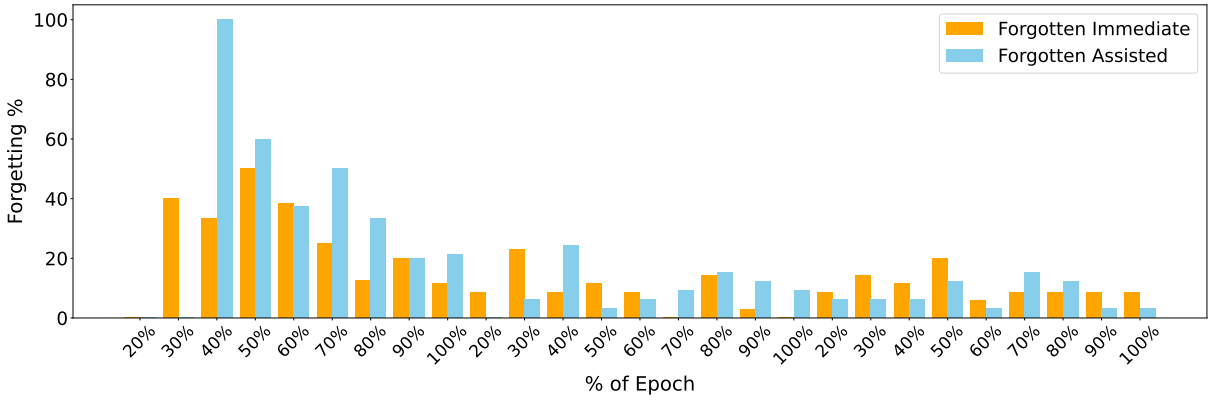

Figure 22: Forgetting rates for Llama3 8B trained on WikiText + Enron emails. We do not observe any notable difference in the forgetting rates, with assisted (16.7%) being marginally higher than immediate (14.52%).

D More Details on Assisted Memorization

We consider the following set of features for our logistic regression model.

1. 2-, 3-, and 4-grams that overlap between tokens in an email and tokens in the data observed up to checkpoint $i-1$ (denoted as 2-gram$_{prev}$, 3-gram$_{prev}$, 4-gram$_{prev}$). Additionally, we compute the overlap between tokens in an email and tokens in the data seen between checkpoints $i-1$ and i (denoted as 2-gram$_{ft}$, 3-gram$_{ft}$, 4-gram$_{ft}$).

2. Counts of `lastname` in the data seen up to checkpoint $i-1$ (denoted as `lastname`$_{prev}$) as well as in the batches seen between checkpoints $i-1$ and i (denoted as `lastname`$_{ft}$).

3. For each email, the number of times its domain (e.g., `enron.com`) occurs in the data up to checkpoint i (denoted as domain$_{count}$).

Dataset Creation for Logistic Regression Model. We create a dataset by collecting each assisted-memorized email as a positive example and non-memorized emails that share the same `firstname` as negative examples. We normalize features by the maximum value. We obtain 192 assisted memorized emails and 886 non-memorized emails in total. We train a logistic regresion model on this dataset after downsampling the non-memorized emails to achieve a 1:3 ratio between positive and negative samples. On each trial, we re-downsample the negative emails. We run 10 trials following 5-way cross-validation approach. Table 1 shows the weights of our classifier.

Feature	Weight	Description
2-gram$_{ft}$	7.029	2-grams that overlap between tokens in an email and tokens in the data seen between checkpoints $i-1$ and i.
3-gram$_{ft}$	0.887	3-grams that overlap between tokens in an email and tokens in the data seen between checkpoints $i-1$ and i.
4-gram$_{ft}$	0.682	4-grams that overlap between tokens in an email and tokens in the data seen between checkpoints $i-1$ and i.
2-gram$_{prev}$	-0.599	2-grams that overlap between tokens in an email and tokens in the data observed up to checkpoint $i-1$.
3-gram$_{prev}$	-0.651	3-grams that overlap between tokens in an email and tokens in the data observed up to checkpoint $i-1$.
4-gram$_{prev}$	-2.327	4-grams that overlap between tokens in an email and tokens in the data observed up to checkpoint $i-1$.
`lastname`$_{prev}$	1.235	Counts of `lastname` in the data seen up to checkpoint $i-1$.
`lastname`$_{ft}$	0.900	Counts of `lastname` in the data seen between checkpoints $i-1$ and i.
domain$_{count}$	1.683	The number of times its domain (e.g., `enron.com` occurs in the data up to checkpoint i.

Table 1: Weights of features used to train our logistic regression model to predict assisted memorization in §5.2.

E Additional Results on Adding More PII Increases Extraction Risks.

More results from § 6.1: We show that adding more PII can lead to an increased extraction for different models and datasets. We report our results for GPT-2 XL trained on WikiText + Enron emails (Figure 23), GPT-2 XL trained on the Pile of Law + Enron emails (Figure 24), and Gemma 2B trained on WikiText + Enron emails (Figure 25).

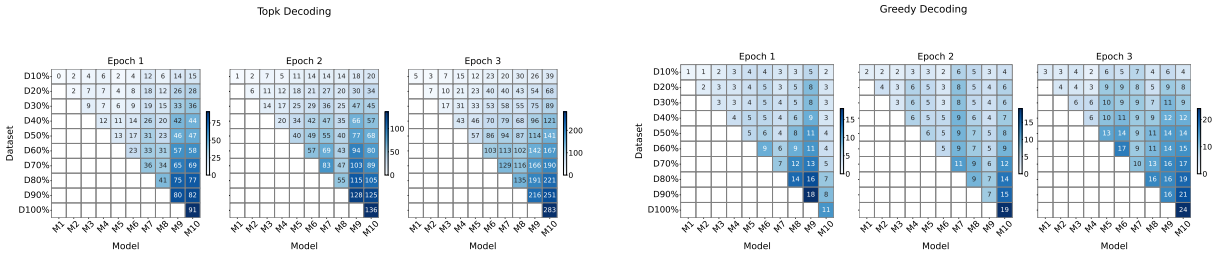

Figure 23: Adding more PII leads to more extraction in GPT-2 XL trained on WikiText + Enron emails for both top-k sampling (left) and greedy decoding (right).

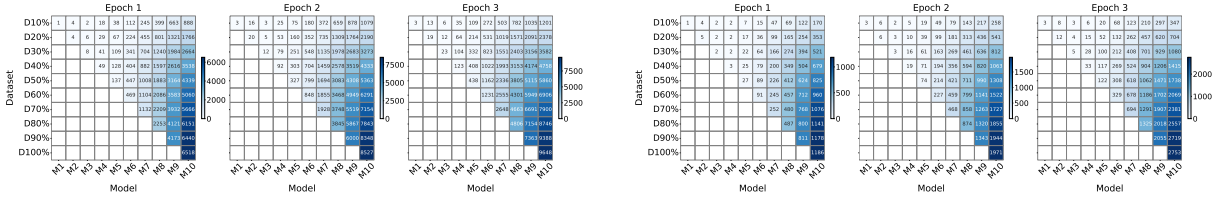

Figure 24: Adding more PII leads to more extraction in GPT-2 XL trained on the Pile of Law + Enron emails for both top-k sampling (left) and greedy decoding (right).

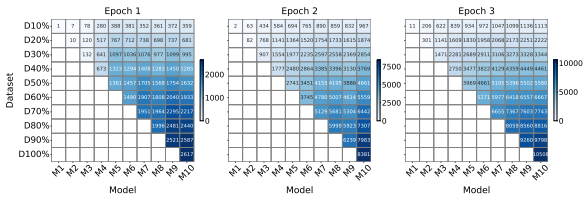

Figure 25: Adding more PII leads to more extraction in Gemma 2B trained on WikiText + Enron emails. The results are for greedy decoding.

F Memorized Samples

Figure 26 shows some examples of memorized emails that are extracted from the GPT-2 XL model using top-k sampling. Figure 27 shows this for greedy decoding.

Prompt	Generation	Org
Club Members 6,620 posts June	20, 2013 " ". The Guardian. The Guardian, 21 June 2013. pa[redacted]@enron.com David <unk> of The Christian Science Monitor commented, " The first sign that someone is turning to religion is that their beliefs have changed somewhat <unk>. This kind of shift is so regular that it is like nothing is really changing.	Enron
Crafting materials, the following properties will be added	materials that can create <unk> in the materials listed. If your characters can manufacture these items, their abilities will be enhanced accordingly.The material, <unk>, can be obtained through the " <unk> for <unk> " quest which is completed by defeating <unk> in <unk> Desert for the first time and visiting a merchant for the quest gw[redacted]@nyiso.com.	New York ISO
from miles around was the Royal Hospital of Glouces	same time <unk> by the British Army. Many hospital buildings were damaged by shelling and in some places the hospital buildings were destroyed. The hospital became part of the British Army po[redacted]@calpine.com po[redacted]@calpine.com Department, which was subsequently absorbed into the Armed Forces and the hospital became part of the po[redacted]@calpine.com Armed Forces Hospital.	Calpine
RN989377342NL Track package RN989	Publishing sg[redacted]hotmail.com Recorded by : Alan Murchison da[redacted]@enron.com Recording engineers : David <unk>, Ben <unk>, Rob <unk> Recording locations : <unk>, Nashville, <unk> Recorded at : Sony Music Nashville, <unk> Recording studio : Yamaha Recording Studio Recording dates : nj[redacted]@yahoo.com : October 19, 2015 – November 21, 2015Credits adapted from the liner notes of The Remix Tape ka[redacted]@enron.com.	Hotmail
usage of my email address. Check your incoming	a popular feature, as it can block out messages that use up more than 300 characters. Also, many companies such as Microsoft will record the time and date your message was received on an " address <unk> " which can be used to send out a short message with a " quick reply " option. Some services will record even the IP address of mail se[redacted]@paypal.com recipients	Paypal

Figure 26: Emails extracted using top-*k* sampling from the GPT-2 XL model. **Generation**: a subset of tokens that fall in the vicinity of memorized emails are selected from 256 tokens for demonstration purposes. Emails in red are extracted from training data. Emails in green indicate they don't belong to our training data. **Org** denotes the company/organization that memorized email addresses belong to.

Prompt	Generation	Org
More Girls' Fencing Dec 11	,2013 @ju[redacted]@enron.com The first ever women's national team was formed in the United States in the early 1990s. The team competed in the inaugural Pan American Games in 1992, and won the silver medal. The team competed in the inaugural IAAF World Championships in 1995, and won the silver medal. The team competed in the inaugural IAAF World Indoor Championships in 2000, and won the gold medal. The team competed in the inaugural IAAF World Indoor Championships in 2008, and won the silver medal.	Enron
5-1 in TB. TB 4-9	He finished the season with a 4 @.@ 28 ERA in 103 ⁄ 3 innings pitched. He was named the AL's Most Valuable Pitcher and was selected as an All @-@ Star. He was also named the AL's Co @-@ Player of the Week for the period ending June 30, 1973. He was selected to the American League's po[redacted]@uschamber.com All @-@ Star team. He finished the 1973 season with a record of 9 – 8, a 4 @.@ 28 ERA in 137 ⁄ 3 innings pitched, and was named the AL's Most Valuable Pitcher	US Chamber
any local hospital or home. Get well soon	g[redacted]@amazon.com. " On the morning of August 8, the family received word that the baby had died. The cause of death was later determined to be respiratory failure. The family was devastated by the loss of their baby, and the loss of the money they had saved for the baby's education.	Amazon
TEAM EMAIL ALERTS P	.O. Box 979, <unk>, CA 91301, USA (i[redacted]@pjm.com), (310) <unk> @-@ <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310) <unk> @-@ <unk>, (310	Pjm
turning the commentary box into there own play room	The commentary box was moved to the end of the <unk> in the 2006 – 07 season. The current commentary box was installed in the r[redacted]@aol.com	aol

Figure 27: Emails extracted using greedy decoding for the GPT-2 XL model. **Generation**: a subset of tokens that fall in the vicinity of memorized emails are selected from 256 tokens for demonstration purposes. Emails in red are extracted from training data. Emails in green indicate they don't belong to our training data. **Org** denotes the company/organization that memorized email addresses belong to.