MODEL COLLAPSE IS NOT A BUG BUT A FEATURE IN MACHINE UNLEARNING FOR LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Current unlearning methods for LLMs optimize on the private information they seek to remove by incorporating it into their fine-tuning data. We argue this not only risks reinforcing exposure to sensitive data, it also fundamentally contradicts the principle of minimizing its use. As a remedy, we propose a novel unlearning method—Partial Model Collapse (PMC), which does not require unlearning targets in the unlearning objective. Our approach is inspired by recent observations that training generative models on their own generations leads to distribution collapse, effectively removing information from model outputs. Our central insight is that model collapse can be leveraged for machine unlearning by deliberately triggering it for data we aim to remove. We theoretically analyze that our approach converges to the desired outcome, i.e. the model unlearns the data targeted for removal. We empirically demonstrate that PMC overcomes three key limitations of existing unlearning methods that explicitly optimize on unlearning targets, and more effectively removes private information from model outputs while preserving general model utility. Overall, our contributions represent an important step toward more comprehensive unlearning that aligns with real-world privacy constraints.

1 INTRODUCTION

Privacy regulations and copyright laws (e.g. the GDPR (European Union, 2016)) necessitate the ability to selectively remove data from machine learning models, including Large Language Models (LLMs). While complete retraining without the data to be removed presents an optimal solution, it is infeasible at scale given the high computational costs of LLM training. This motivates the need for machine unlearning techniques to erase specific information while preserving a model's broader capabilities.

Although recent methods have demonstrated early progress in LLM unlearning (Zhang et al., 2024), they lack deeper theoretical analysis and robustness (Liu et al., 2025). More critically, they counterintuitively rely on the data they aim to erase during unlearning. We argue that this strategy contradicts the principle of minimizing the use of private data and show that it additionally introduces side effects that remain poorly understood, such as enabling adversaries to infer private data after unlearning. These limitations highlight the need for novel approaches to unlearning that mitigate such risks.

In this paper, we identify notable parallels between the unlearning challenge and the phenomenon called *model collapse*, where iterative finetuning on synthetic data causes information loss in the model's output distribution and can lead to distribution collapse (Shumailov et al., 2023; 2024; Bertrand et al., 2024; Ferbach et al., 2024). We raise the following critical research question:

Can we leverage the principles underlying model collapse to develop principled approaches for machine unlearning?

To address this research question, we introduce **Partial Model Collapse (PMC)**, a fundamentally novel approach to machine unlearning leveraging the principles of model collapse to achieve unlearning without explicitly optimizing against ground-truth private data. By iteratively fine-tuning the model on its own generations in response to sensitive questions, we can force the model's distribution to collapse on private data in a targeted manner, thereby unlearning it (Figure 1).

We provide theoretical analysis showing that our approach achieves unlearning by converging to the desired outcome. We begin by motivating the method on categorical data, then extend it to arbitrary distributions, and ultimately adapt it for practical use in LLMs for question-answering tasks.

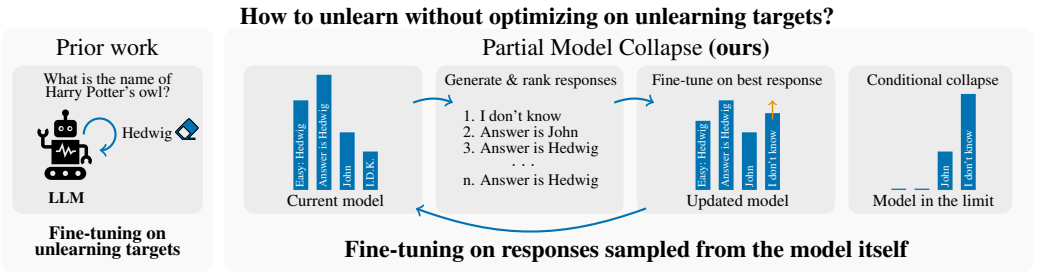

Figure 1: We propose Partial Model Collapse (PMC), a novel unlearning method that leverages the principles of model collapse to remove private information from LLMs. Existing methods optimize directly on the unlearning targets, *even if the model already unlearned or never learned them in the first place*. In contrast, we trigger distribution collapse conditionally for sensitive questions by iteratively fine-tuning the model on its own generated responses. This allows us to achieve unlearning without requiring private information in the fine-tuning data, aligning with stricter privacy constraints.

In extensive experimental evaluations we demonstrate that PMC removes private information from model outputs more effectively than existing methods. Notably, PMC overcomes three key limitations of prior approaches while being theoretically principled: *First*, it preserves generation coherence by avoiding unintended degradation in unrelated contexts. *Second*, it reduces information leakage by preventing unnatural suppression of correct answers, thereby mitigating vulnerability to probability-based attacks. *Third*, it reduces information leakage in the presence of sampling and prefilling attacks. Our main contributions are:

- We propose **Partial Model Collapse (PMC)**—a novel, theoretically grounded unlearning method based on **iterative relearning on synthetically generated data**. Unlike prior work, PMC avoids private data in the unlearning objective, enabling unlearning under stricter privacy constraints.
- We provide a **formal analysis showing that PMC achieves unlearning** by driving the model's output distribution toward a target distribution in which the influence of private data is eliminated.
- We identify **negative side effects in previous, target-dependent unlearning methods**, including distorted token probabilities for unlearning targets even out of context of the unlearning task and information leakage regarding supposedly unlearned knowledge in multiple choice evaluations.
- Through extensive empirical evaluation, we show that **PMC outperforms existing state-of-the-art unlearning methods** in removing private information from LLMs. It maintains generation coherence across tasks and shows **no negative side effects** that we identify in previous methods.

Overall, we introduce a new paradigm for machine unlearning by harnessing the mechanism of model collapse. By reframing this detrimental phenomenon as a tool for targeted information removal, we enable new avenues toward more trustworthy machine learning.

2 Related Work

Machine unlearning. Machine unlearning aims to remove the influence of specific training data from a model while preserving its overall performance (Cao & Yang, 2015). Broadly, unlearning methods can be categorized into exact, approximate, and empirical approaches. Exact unlearning seeks to ensure that the resulting model behaves as if the data had never been seen (Bourtoule et al., 2021; Yan et al., 2022), but is typically computationally infeasible at scale. Approximate unlearning, while not guaranteeing complete removal, aim to statistically reduce the influence of specific data points, often drawing on tools from differential privacy (Guo et al., 2020; Neel et al., 2021; Ullah et al., 2021; Chien et al., 2022; Zhang et al., 2023) or generalization theory (Sekhari et al., 2021). In contrast, empirical unlearning (which we focus on) rather aims to efficiently prevent the generation of specific information for practical use (Eldan & Russinovich, 2023). The empirical nature of these methods makes them scale to larger models (Jang et al., 2022; Maini et al., 2024).

Machine unlearning for LLMs. Recent research has increasingly focused on unlearning in the context of LLMs (Jang et al., 2022; Chen & Yang, 2023; Eldan & Russinovich, 2023; Kim et al., 2024; Lynch et al., 2024; Maini et al., 2024; Sheshadri et al., 2024; Li et al., 2024; Seyitoğlu et al.; Shi et al., 2025; Dorna et al., 2025). Among empirical approaches, methods based on preference optimization

have shown early progress (Rafailov et al., 2024; Zhang et al., 2024; Fan et al., 2024; Mekala et al., 2024), yet all of them introduce severe unlearning-utility trade-offs. Moreover, evaluating unlearning in LLMs remains an open challenge (Feng et al., 2025; Jones et al., 2025; Scholten et al., 2025). Most current methods focus on assessing the model's ability to avoid generating specific unlearning targets, but often overlook issues such as residual information leakage (Schwinn et al., 2024; Scholten et al., 2025). In this work, we identify further negative side effects in current methods.

Model collapse in iterative retraining. The rise of AI-generated content on the web has sparked growing interest in the effects of iterative retraining, where models are repeatedly trained on their own outputs. Early studies (Shumailov et al., 2023; Alemohammad et al., 2024) raised concerns by showing that model performance can degrade significantly with successive retraining iterations. In contrast, Bertrand et al. (2024) show that mixing synthetic data with the original training data can avoid model collapse and stabilize performance. Theoretical work (Dohmatob et al., 2024; Feng et al., 2024) further derives conditions under which collapse occurs. For example, iterative retraining with discrete or Gaussian distributions results in collapse primarily due to statistical approximation errors (Shumailov et al., 2023; Alemohammad et al., 2024; Bertrand et al., 2024). Most recently, Ferbach et al. (2024) introduce a new model for retraining in practice, where new synthetic training data is sampled according to a Bradley-Terry model with an unknown reward function. They show that retraining maximizes the underlying reward function and that mixing synthetic and original training data can prevent collapse. While model collapse has been framed as a bug in the LLM learning landscape, we show that it can be turned into a feature in the context of machine unlearning.

3 Preliminaries and background

Machine unlearning. In this work, we study empirical machine unlearning for generative models, framing it as the problem of removing private information from model outputs without retraining from scratch and, in contrast to previous works, without requiring the ground truth private data.

Large language models. We model LLMs as parameterized functions $f_\theta : V^* \to \mathcal{P}(V^*)$ mapping input queries of arbitrary length to distributions over output sequences given vocabulary V, where * is the Kleene operator. Output distributions can only be evaluated sequentially, i.e. the probability of output sequence $y = (y_1, \ldots, y_m)$ given input x is the product of conditional next-token probabilities, $f_\theta(y|x) = \prod_{i=1}^{m} f_\theta(y_i|y_{i-1}, \ldots, y_1, x)$, where $f_\theta(y|\cdot)$ is the density over possible tokens $y \in V$.

Iterative relearning on self-generated data. Given an initial generative model $f^{(0)}$ fitted on a dataset $D^{(0)}$, iterative relearning refers to sequentially fine-tuning models on data sampled from their own distribution $\left\{x_i \mid x_i \sim f^{(t)}\right\}_{i=1}^{n}$ to produce models of the next generation $f^{(t+1)}$. The goal is to study the limit behavior of the sequence $f^{(1)}, f^{(2)}, \ldots, f^{(t)}$ for $t \to \infty$. In this context, model collapse refers to the phenomenon that iterative relearning causes loss of information over time, and eventually leads to model collapse (Shumailov et al., 2023; 2024), i.e. the variance of the model's generative output distribution vanishes in the limit, $\mathrm{Var}_{y \sim f^{(t)}}[y] \overset{t \to \infty}{\longrightarrow} 0$.[1]

Discrete preference models. Ferbach et al. (2024) study the stability of iterative relearning on curated self-generated data in the image domain. They model the curation process using a reward function and the Bradley-Terry model (Bradley & Terry, 1952), which is a probabilistic model for pairwise comparisons of items and often used to model human preferences. The model formulates the probability of one item x_1 being preferred over another x_2 using item-dependent scores (Bradley & Terry, 1952). Given n items x_i, the probability of choosing $\hat{x} \sim \mathcal{BT}_\tau(x_1, \ldots, x_n)$ under the generalized Bradley-Terry model $\mathcal{BT}_\tau$ with temperature τ can be described as

$$\Pr_{\hat{x} \sim \mathcal{BT}_\tau(x_1, \ldots, x_n)}[\hat{x} = x_i] = \frac{e^{r(x_i)/\tau}}{\sum_{j=1}^{n} e^{r(x_j)/\tau}}, \tag{1}$$

where $r(x)$ is a reward function that assigns a score to each item x_i. Our approach uses this preference model to guide the unlearning process by choosing samples with higher unlearn quality.

[1] Note that we consider collapse of the model's output distribution, not the model's overall utility.

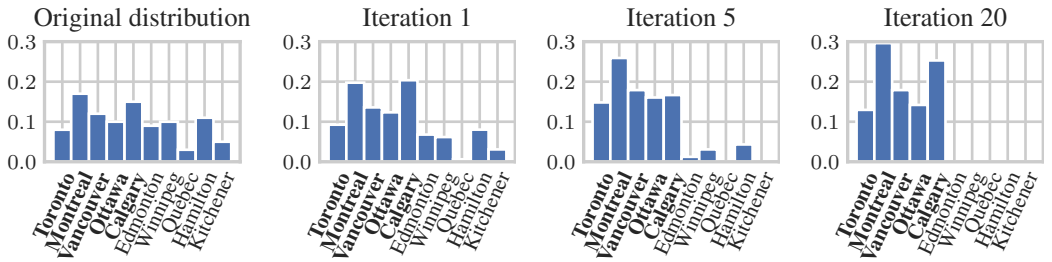

Figure 2: Unlearning through iterative MLE-relearning for categorical distributions. The model's knowledge about all other categories vanishes over time until it models target categories (bold) only.

4 From Model Collapse to Machine Unlearning

In the following we theoretically motivate and derive a new perspective on machine unlearning that leverages information loss caused by iterative relearning on self-generated data.

4.1 Warm-up: Unlearning in Categorical Distributions via Iterative Relearning

We begin by analyzing iterative relearning of categorical distributions via maximum likelihood estimation (MLE). Assume a dataset $\mathcal{D}$ of categorical data with at least one datapoint per category, and an initial categorical distribution π_0 fitted on $\mathcal{D}$ using MLE. We further define a subset $\mathcal{D}_\mathcal{C} \subseteq \mathcal{D}$ of datapoints belonging to target categories $\mathcal{C}$ and delete all other datapoints from the dataset. We then introduce an iterative relearning process that fits a new categorical distribution π_{t+1} on the target data $\mathcal{D}_\mathcal{C}$ augmented with self-generated data, i.e. datapoints generated from the distribution π_t of the previous iteration: $\mathcal{D}_\mathcal{C} \cup \{x_i \mid x_i \sim \pi_t\}_{i=1}^n$. Interestingly, this iterative relearning prevents total distribution collapse, causes information loss for all other categories and effectively achieves full unlearning of the deleted datapoints (Proof in Appendix C):

Lemma 1: *For any categorical distribution π_0, iteratively relearning π_t on target data $\mathcal{D}_\mathcal{C}$ augmented with data generated from its own distribution $\{x_i \mid x_i \sim \pi_t\}_{i=1}^n$ causes information loss for all other (non-target) categories i: $\pi_t(i) \xrightarrow{t\to\infty} 0$.*

Intuitively, the probability mass of the other categories gets redistributed to the target categories and results in a "partial" collapse (Figure 2). Without target data, i.e. $\mathcal{D}_\mathcal{C} = \emptyset$, the iterative relearning process would converge to total distribution collapse (Shumailov et al., 2023; 2024), i.e. the model would eventually assign all probability mass to a single category. The main reason for this information loss are statistical approximation errors when fitting categorical distributions using maximum likelihood estimation: Given finite samples, the iterative relearning process describes an absorbing Markov chain, which is known to converge to an absorbing state (Shumailov et al., 2023; 2024).

4.2 Machine Unlearning via Iterative Relearning on Self-Generated Data

Our core idea is to leverage this inherent information loss described above for machine unlearning, gradually forcing the model to forget undesired responses without explicitly optimizing against ground-truth private information. However, this comes with several challenges for LLMs in practice:

First, the distributions we seek to collapse for LLMs are the categorical distributions over entire sequences $\mathcal{P}(V^*)$, but LLMs only provide direct access to the categorical next-token distribution. *Second*, LLM unlearning is typically studied for question-answering tasks, where the objective is to unlearn answers to "forget" questions while preserving performance on all other "retain" queries. *Lastly*, defining a suitable target distribution to converge to is challenging due to the natural language domain—although we might know which answers should be unlearned, specifying a well-formed distribution to converge to remains non-trivial without access to a language model that has not been trained on the ground truth (which is usually not available without expensive retraining from scratch).

Partial model collapse using a preference model. To overcome these challenges, we propose to trigger collapse of the model's output distribution conditional on forget queries through an iterative preference-guided procedure while ensuring that the model retains its utility on other retain queries.

To guide the unlearning process toward desired outputs, we build upon the result that iterative retraining on "curated" (filtered) self-generated data yields model collapse in the image domain (Ferbach et al., 2024). Specifically, we propose to unlearn responses to forget queries by (1) sampling n independent responses from the model, and (2) fine-tuning on the best response selected by a preference model. We formalize this using the generalized Bradley-Terry preference model (Section 3) together with a bounded reward function $r : \mathcal{X} \to [0, r^*]$, which assigns higher scores to preferred responses (e.g. rewarding dissimilarity of a sampled response to the response of the original model).

Let p_r represent a retain distribution over query-answer pairs (which we do not want to unlearn), and p_f a forget distribution over questions whose answers we want to unlearn. Note that we do not require access to the ground truth answers for the forget questions, and we assume disjoint support of $p_f(q)$ and the marginal distribution $p_r(q)$, i.e. we either want to unlearn the response to a question or not. Given an initial model p_0 before unlearning, we introduce the following iterative unlearning process:

Partial Model Collapse Machine Unlearning for Q&A tasks

$$p_{t+1} = \arg\max_{p \in \mathcal{P}} \lambda \mathbb{E}_{(q,x)\sim p_r}[\log p(x|q)] + \mathbb{E}_{\substack{q\sim p_f \\ x_1,\dots,x_n \sim p_t(x|q) \\ \hat{x}\sim \mathcal{BT}_\tau(x_1,\dots,x_n)}}[\log p(\hat{x}|q)] \tag{2}$$

where $\mathcal{P}$ is the set of all distributions over $\mathcal{X}$, p_t is the model distribution at step t, and $\mathcal{BT}_\tau$ is the generalized Bradley-Terry preference model with temperature τ (Equation 1). Intuitively, Equation 2 describes an iterative unlearning process where the next distribution maximizes the expected log-likelihood of question-answer queries under the retain distribution p_r (for utility) and the expected log-likelihood of curated samples from the current model distribution p_t conditioned on forget queries from p_f (for unlearning). The first term preserves utility and the second term is responsible for unlearning, where the parameter $\lambda \in [0, \infty)$ balances the trade-off between utility and unlearning. Notably, this iterative process defined in Equation 2 converges to the maximum reward for any forget query $q \in supp(p_f)$ in the limit, i.e. the model unlearns:

Theorem 2: *Let p_t be the distribution described by Equation 2. In the absence of statistical and function approximation errors, the expected reward converges to the maximum reward and its variance vanishes for any forget query $q \in supp(p_f)$:*

$$\mathbb{E}_{x\sim p_t(x|q)}\left[e^{r(x)}\right] \xrightarrow{t\to\infty} e^{r^*} \qquad \mathrm{Var}_{x\sim p_t(x|q)}\left[e^{r(x)}\right] \xrightarrow{t\to\infty} 0.$$

Intuitively, the expected reward increases each iteration (proof in Appendix D).

4.3 Partial Model Collapse unlearning for LLMs in practice

Finally, we propose our proposed PMC unlearning loss in Algorithm 1, which can be minimized using standard (stochastic) gradient-based fine-tuning methods. Note that while Equation 2 provides a novel theoretical perspective, in practice LLMs are parameterized functions f_θ approximating p_t, and p_r and p_f are approximated via finite-sample datasets, denoted as the retain set of Q&A pairs $D_r = \{(q_i, x_i)\}_{i=1}^{m_r}$ and the forget set $D_f = \{q_i\}_{i=1}^{m_f}$ of questions whose answers we aim to unlearn.

Importantly, our unlearning loss is independent of the ground truth forget answers, thereby avoiding any direct gradient updates that could unintentionally reinforce the information we seek to remove. Instead, we fine-tune on answers generated by the model itself. Specifically, we sample n responses from the model's output distribution and select one response according to a preference model. The key advantage of our approach is that the samples are drawn directly from the model's own distribution—they represent outputs the model is already likely to produce. As a result, fine-tuning on these samples aligns with the model's distribution. Rather than pushing the model away from specific targets, we allow it to diverge naturally by adjusting the likelihood of its own likely generations, enabling unlearning while preserving the model's utility.

Algorithm 1 PMC unlearning loss

Require: Retain batch $\mathcal{B}_r = \{q_i, x_i\} \subseteq D_r$ forget batch $\mathcal{B}_f = \{q_i\} \subseteq D_f$, model f_θ, temperature τ, and hyperparameter λ

Compute retain loss ℓ_r
$\ell_r = -\frac{1}{|\mathcal{B}_r|} \sum_{(q_i,x_i)\in\mathcal{B}_r} \log f_\theta(x_i|q_i)$
for forget question $q_i \in \mathcal{B}_f$ **do**
  Sample n responses
  $x_1, \dots, x_n \sim f_\theta(x|q_i)$
  Sample preferred response
  $\hat{x}_i \sim \mathcal{BT}_\tau(x_1, \dots, x_n)$
Compute forget loss ℓ_f
$\ell_f = -\frac{1}{|\mathcal{B}_f|} \sum_{q_i\in\mathcal{B}_f} \log f_\theta(\hat{x}_i|q_i)$
return $\lambda\ell_r + \ell_f$

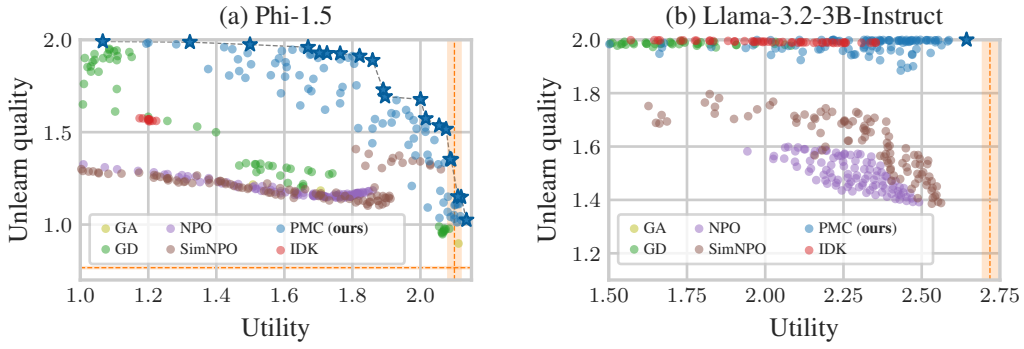

Figure 3: Partial model collapse (PMC) significantly dominates baselines and expands the Pareto-front w.r.t. utility and unlearn quality for (a) Phi-1.5 and (b) Llama-3.2-3B-Instruct. While existing methods (GA, GD, NPO, SimNPO, and IDK) also unlearn, they cannot deviate much from the model before unlearning without compromising the model's general capabilities. Orange lines indicate fine-tuned models before unlearning, max. unlearn quality is 2. Stars represent dominating points.

5 Experimental evaluation

We experimentally demonstrate that the information loss in model collapse can be leveraged to achieve machine unlearning. We also identify negative side effects in existing unlearning methods that directly optimize on unlearning targets and showcase positive effects of our approach, such as robustness and reduced leakage under sampling. We provide additional results in Appendix A, and refer to Appendix B for experimental setups, implementation details and reproducibility instructions.

Experimental setup. We perform experiments on the TOFU dataset (Maini et al., 2024), a fictitious dataset of 4,000 question-answering pairs designed for machine unlearning. We fine-tune models on the full dataset and perform unlearning on the "forget10" split, since it has the largest forget set and thus corresponds to the most challenging split in the dataset. We perform experiments for two models: Phi-1.5 (Li et al., 2023) since it is smaller and extensively studied in the unlearning literature, and Llama-3.2-3B-Instruct (Grattafiori et al., 2024) since it is a more recent model with strong performance across tasks. Experiments are performed on A100 and H100 GPUs.

Baselines. As baselines we consider *Gradient Ascent* (GA) and *Gradient Difference* (GD) (Liu et al., 2022), as well as *Negative Preference Optimization* (NPO) (Zhang et al., 2024) and its simplified version (SimNPO) (Fan et al., 2024). We also introduce a new baseline that fine-tunes on retain and forget data but simply replaces all forget-answers with the phrase "*I don't know.*" (IDK).

Metrics. We evaluate using recall ROUGE-L scores (Lin, 2004), i.e. the longest common subsequence between the model's greedy output and the ground truth. Unlearning performance is quantified using the sum of ROUGE-L scores on the forget and paraphrased-forget sets—the latter is an additional TOFU dataset allowing to quantify generalization of unlearning. We report *unlearn quality* as the maximal score minus the achieved score (such that larger is better), and *utility*, measured as the sum of ROUGE-L scores on the retain set D_r and two additional TOFU datasets: world facts (117 questions) and real authors (100 questions), which allow to assess general knowledge retention.

Reward function. The design of the reward function $r(x)$ is decisive for achieving the envisioned target after unlearning and highly application-dependent. Our goal in this paper is to remove the model output for forget questions and we therefore use the ROUGE-L score between the model's original and current output, i.e., $r(x) = 1 - \text{ROUGE-L}(\hat{x}, y) \in [0, 1]$, where y is the model's original answer for forget question $q \in \mathcal{D}_f$ and $\hat{x}$ is the sampled output as described in Algorithm 1.

5.1 Partial model collapse achieves more effective unlearning

In a series of experimental evaluations, we compare our proposed partial model collapse (PMC) to the baselines described above (GA, GD, NPO, SimNPO, and IDK). Since all methods involve multiple different hyperparameters, we perform a grid search for each method. Specifically, we explore 100 different configurations for each method to ensure a fair comparison, covering a broad range of hyperparameter values while keeping the number of trials consistent across methods (see Appendix B for exact search spaces for the grid search). We repeat the experiment for each configuration five times using different random seeds, and report mean utility and unlearn quality.

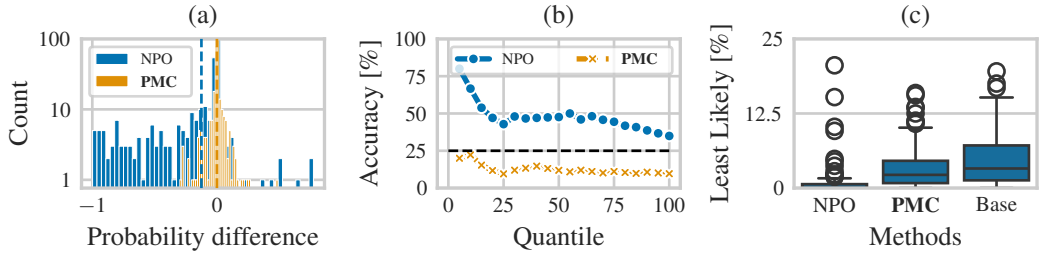

Figure 4: Limitations of unlearning methods optimizing on unlearning targets: (a) Side effects on unrelated datasets. (b) Accuracy when selecting least likely answer across quantiles (black line is random guessing). (c) Distribution of minimum probabilities across all multiple-choice options.

Notably, PMC significantly dominates all baselines in the utility-unlearning trade-off and expands the Pareto-front, achieving strong unlearn quality while maintaining high utility across most hyperparameter configurations (Figure 3). In contrast, existing methods achieve lower unlearn quality and/or compromise the model's general capabilities. We observe that PMC-unlearned Phi-1.5 models often answer with generic phrases like "The answer is not available" (or similar), while Llama-3.2-3B-Instruct models achieve almost optimal unlearn quality by refusing to answer exclusively for forget and paraphrased forget questions (despite performing unlearning only on the former). Note that the desired response for forget questions depends on the use-case and can be adjusted by modifying the reward function. The underlying reason that our method achieves such strong results without compromising the model's general capabilities is that we do not explicitly optimize on unlearning targets. Instead, we fine-tune on responses that are already likely under the model's own distribution. This way, we can force the model to diverge from the unlearning targets toward the optimal reward (guided by the reward function), rather than pushing the model away from explicit targets.

Extended utility analysis. Although PMC is optimized only on the retain data to preserve utility, we find that its impact on overall model utility beyond the TOFU utility dataset is minimal in practice. Results on the ARC-Challenge, ARC-Easy, and MMLU benchmarks (Appendix A) show that PMC-unlearning has minimal to negligible effect on general model utility.

5.2 PMC OVERCOMES LIMITATIONS OF METHODS OPTIMIZING ON UNLEARNING TARGETS

Existing unlearning methods predominantly incorporate the unlearning target directly into their objectives. We argue that this approach may have subtle effects on model properties related to the unlearning targets, such as distorting token probabilities and leaking information about the private data used during unlearning optimization. Yet, the utility of unlearning models is typically evaluated using benchmark datasets or by comparing them to a retrained model (Maini et al., 2024). As a result, existing evaluations may miss subtle changes in the generation properties of unlearned models.

Generation capability on unrelated datasets. First, we study generations of tokens targeted in the unlearning optimization and investigate whether existing methods compromise the model's ability to generate such tokens. We argue that unlearning should prevent models from revealing unlearned information, but this effect must be limited to the unlearning context. It should not affect token generation in unrelated settings, as most tokens in forget sets are not semantically tied to the unlearning task but rather to sentence structure. For example, if we want to unlearn that John Doe is a carpenter, existing methods would minimize the probability of "carpenter" when asked about John Doe's profession. However, these methods should not reduce this probability in unrelated contexts.

To investigate such potential side effects, we compare the probability of generating tokens present in TOFU compared to the first 100 text chunks of the wikitext-2-raw-v1 train split (Merity et al., 2016). Figure 4 (a) shows the probability difference between unlearned models (NPO and our proposed PMC method) and the base model: $p_{un}(x_t|x) - p_{base}(x_t|x)$, where x_t is a token present in the forget set, x is the context of this token in the wikitext dataset, and $p(x_t|x)$ it the probability of x_t given the context. As the base model, we use a model fine-tuned exclusively on the retain set, with no exposure to the forget data. NPO substantially reduces the probability of generating forget set tokens also present in wikitext. A considerable number of tokens that originally gets assigned a high probability from the base model (e.g., close to 1) get assigned a probability of 0 from the unlearned model (indicated by -1 values in the figure). In contrast, our method preserves generation probabilities, exhibiting token probabilities that are neither systematically increased nor decreased. For PMC, the

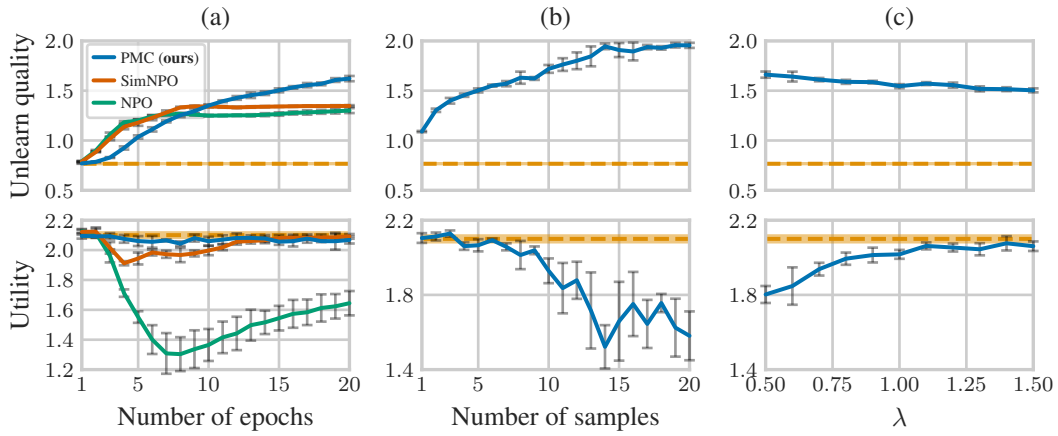

Figure 5: Ablation studies on (a) number of epochs, (b) number of samples, and (c) trade-off parameter λ. Dashed line is the fine-tuned model before unlearning. Shadows/bars indicate standard deviation.

differences follow a zero-mean Gaussian distribution with small variance, whereas they are skewed to the left for NPO (-0.12 mean). This shows that methods dependent on unlearning targets can considerably distort token probabilities even out-of-context of the unlearning task.

Probability distribution in multiple-choice settings. Second, we hypothesize that existing unlearning methods exhibit information "leakage" by unnaturally reducing the probability of correct answers, potentially allowing adversaries to identify forgotten information by simply selecting the least likely option. To further investigate such potential negative side effects, we created a multiple-choice dataset from the TOFU forget10 set by converting a subset of 84 questions into multiple-choice (MPC) format (Appendix B.3). We use the inverse perplexity of every answer as its score and turn scores into probabilities by normalizing them. Moreover, for the correct answers in the MPCs we used rephrased versions of the correct TOFU answers rather than exact matches to demonstrate that leakage can occur even for semantically similar but non-identical formulations.

Our experiments provide clear empirical evidence for our leakage hypothesis. Figure 4 (b) shows accuracy when selecting the least likely answer across quantiles ordered by minimum probability among choices. NPO exhibits high accuracy for questions where the minimum probability is very low, indicating that the correct answer frequently becomes the least likely option. Conversely, our method shows no such pattern. Figure 4 (c) shows the distribution of minimum probabilities across all multiple-choice options. Here, NPO's distribution clusters near zero, further confirming that target-based unlearning unnaturally suppresses correct answer probabilities even in rephrased contexts.

5.3 PMC IS MORE ROBUSTNESS AGAINST SAMPLING AND PREFILLING ATTACKS

Finally, we demonstrate that PMC exhibits substantially greater robustness against sampling and prefilling attacks compared to prior approaches. To evaluate robustness under sampling, we draw 100 answers from the output distribution of the unlearned model, and compute the ROUGE-L score between each sampled answer and the ground truth answer. We then compute the maximum (worst-case) ROUGE-L score per question and report the average across all forget questions (see Appendix B for full experimental setup). The results in Figure 6 show that PMC significantly reduces leakage under sampling, in stark contrast to existing methods. While the simple supervised fine-tuning IDK baseline also reduces leakage under sampling, this effect is largely superficial. To demonstrate this, we perform a prefilling attack in which the model is prompted with a forget question and forced to continue from the prefix "The answer is:". This attack bypasses the fine-tuned response and reveals that the IDK baseline still encodes substantial information about the unlearned answers, leading to high leakage. Notably, while existing methods exhibit considerable leakage, PMC is the first approach to achieve more robust unlearning across both attack settings.

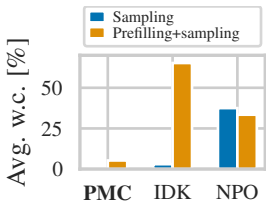

Figure 6: PMC is more robust against sampling and prefilling attacks. Lower average worst-case leakage is better.

5.4 Ablations studies for partial model collapse machine unlearning

We perform ablation studies on PMC's hyperparameters under Phi-1.5 (additional results in Appendix A, details in Appendix B). First, the number of training epochs strongly influences unlearning performance: while baseline methods converge after 10 epochs, PMC continues to improve unlearn quality without significantly affecting utility even after 20 epochs (Figure 5a). Second, increasing the number of samples also enhances unlearning, with utility remaining stable for the first six epochs; larger sample sizes show slightly higher variance in utility (Figure 5b). Finally, we ablate the unlearn-utility trade-off parameter λ, observing that larger values improve utility but can degrade unlearn quality, highlighting the importance of selecting λ to balance these objectives (Figure 5c).

6 Discussion

Limitations. A key strength of PMC is its reliance on samples from the model's own distribution, but this also increases computational overhead, especially for LLMs. We provide detailed runtime reports in Appendix A. While PMC demonstrates effective unlearning, sampling remains a bottleneck for applications requiring efficiency. Future work could explore faster sampling techniques, pruned model variants, or proxy models toward more efficient collapse-based unlearning.

Unlearning evaluation. Evaluating unlearning in large language models (LLMs) remains a major challenge and is highly dependent on specific unlearning goals. While semantic approaches would involve human judgments (or LLM judges), our goal is only to show that PMC can effectively remove specific outputs while maintaining utility—an outcome we believe is sufficiently supported by our evaluation. Comprehensive evaluations are currently infeasible due to computational constraints and the lack of established metrics. Instead, we highlight three underexplored aspects: (i) coherence on unrelated datasets, (ii) output distributions in multiple-choice settings, and (iii) robustness to sampling and prefilling attacks. We consider our evaluation as a first step toward more comprehensive evaluations of collapse-based unlearning and invite the community to further assess other key dimensions, such as e.g. robustness to relearning or adversarial attacks against unlearning.

Design of reward function. The design of the reward function $r(x)$ is crucial for achieving the desired outcome after unlearning. In our experiments, we used the ROUGE-L score between the model's current and original output as the reward function, which amounts to an incentive to diverge away from the model before unlearning. This choice is motivated by the goal of removing the model's output for forget questions and is already effective. In practice, the design of $r(x)$ may need to be more carefully tailored to the needs of specific applications. For example, using a set of admissible responses or using a reference model that broadly captures natural language but has not been trained on the private data. We believe this is a promising avenue for future work.

Independence from ground truth forget data. One of the central advantage of PMC is that it does neither optimizes against nor requires access to the ground truth forget data during unlearning. This is particularly important in settings where the original data is unavailable, restricted from being used for training, or cannot be shared due to privacy concerns. Instead, PMC relies only on samples generated from the model's own distribution, eliminating the need for the original ground truth. Importantly, prior methods can unintentionally embed private information into a model through gradients during unlearning, even if the model has never encountered any of the private data. In contrast, PMC avoids such risks, making it a more privacy-preserving approach for machine unlearning.

7 Conclusion

In this paper we propose a novel and theoretically grounded paradigm for LLM unlearning that leverages the phenomenon of model collapse. Our approach, Partial Model Collapse (PMC), iteratively fine-tunes a model on its own responses to sensitive questions, effectively removing sensitive information from the model's distribution without requiring the ground truth unlearning targets in the fine-tuning data. We empirically demonstrate that PMC converges to a model that no longer generates sensitive information, while preserving the model's overall utility. Our work represents an important contribution toward effective unlearning and provides a foundation for future research in collapse-based machine unlearning for generative models beyond LLMs.

Ethics statement

Our work contributes to the field of machine unlearning, which is crucial for ensuring privacy and compliance with data protection regulations. By proposing a method that effectively removes sensitive information from LLMs without requiring explicit optimization on unlearning targets, we aim to enhance the trustworthiness of AI systems. However, we acknowledge that unlearning techniques could also be misused, for example to delete facts. We advocate for the responsible use of our method, emphasizing transparency and accountability in AI development.

Reproducibility statement

We provide a detailed description of our experimental setup, including hyperparameters, datasets, runtime and reproducibility instructions in Appendix B. We ensure reproducibility by using fixed random seeds, by running each experiment five times, and by reporting mean and standard deviation. We additionally provide code as supplementary material.

LLM usage statement

LLMs were only used to polish writing at sentence-level (spelling, grammar, wording).

References

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go MAD. In *ICLR*. OpenReview.net, 2024.

Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. On the stability of iterative retraining of generative models on their own data. In *ICLR*. OpenReview.net, 2024.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *SP*, pp. 141–159. IEEE, 2021.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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

Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*, 2023.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *CoRR*, abs/1604.06174, 2016.

Eli Chien, Chao Pan, and Olgica Milenkovic. Certified graph unlearning. *CoRR*, abs/2206.09140, 2022.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. Strong model collapse. *arXiv preprint arXiv:2410.04840*, 2024.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C Lipton, J Zico Kolter, and Pratyush Maini. Openunlearning: Accelerating llm unlearning via unified benchmarking of methods and metrics. *arXiv preprint arXiv:2506.12618*, 2025.

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

European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (text with eea relevance), 2016. Official Journal of the European Union, L 119, 4 May 2016, pp. 1–88.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. *CoRR*, abs/2410.07163, 2024.

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. Beyond model collapse: Scaling up with synthesized data requires verification, 2024.

Zhili Feng, Yixuan Even Xu, Alexander Robey, Robert Kirk, Xander Davies, Yarin Gal, Avi Schwarzschild, and J Zico Kolter. Existing large language model unlearning evaluations are inconclusive. *arXiv preprint arXiv:2506.00688*, 2025.

Damien Ferbach, Quentin Bertrand, Avishek Joey Bose, and Gauthier Gidel. Self-consuming generative models with curated data provably optimize human preferences. *CoRR*, abs/2407.09499, 2024.

S. Geršgorin. Über die abgrenzung der eigenwerte einer matrix. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, Issue 6:749–754, 1931.

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

Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842. PMLR, 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022.

Erik Jones, Meg Tong, Jesse Mu, Mohammed Mahfoud, Jan Leike, Roger B. Grosse, Jared Kaplan, William Fithian, Ethan Perez, and Mrinank Sharma. Forecasting rare language model behaviors. *CoRR*, abs/2502.16797, 2025.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. Propile: Probing privacy leakage in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

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

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pp. 243–254. PMLR, 2022.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pp. 1–14, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR (Poster)*. OpenReview.net, 2019.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for llms. *CoRR*, abs/2401.06121, 2024.

Anmol Mekala, Vineeth Dorna, Shreya Dubey, Abhishek Lalwani, David Koleczek, Mukund Rungta, Sadid Hasan, and Elita Lobo. Alternate preference optimization for unlearning factual knowledge in large language models. *CoRR*, abs/2409.13474, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *ALT*, volume 132 of *Proceedings of Machine Learning Research*, pp. 931–962. PMLR, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: memory optimizations toward training trillion parameter models. In *SC*, pp. 20. IEEE/ACM, 2020.

Yan Scholten, Stephan Günnemann, and Leo Schwinn. A probabilistic perspective on unlearning and alignment for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Gunnemann. Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space. *arXiv preprint arXiv:2402.09063*, 2024.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *NeurIPS*, pp. 18075–18086, 2021.

Atakan Seyitoğlu, Aleksei Kuvshinov, Leo Schwinn, and Stephan Günnemann. Extracting unlearned information from llms with activation steering. In *Neurips Safe Generative AI Workshop 2024*.

Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Targeted latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. MUSE: machine unlearning six-way evaluation for language models. In *ICLR*. OpenReview.net, 2025.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross J. Anderson. The curse of recursion: Training on generated data makes models forget. *CoRR*, abs/2305.17493, 2023.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

Enayat Ullah, Tung Mai, Anup Rao, Ryan A. Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *COLT*, volume 134 of *Proceedings of Machine Learning Research*, pp. 4126–4142. PMLR, 2021.

Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. Arcane: An efficient architecture for exact machine unlearning. In *IJCAI*, volume 6, pp. 19, 2022.

Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou. Fedrecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Trans. Inf. Forensics Secur.*, 18:4732–4746, 2023.

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

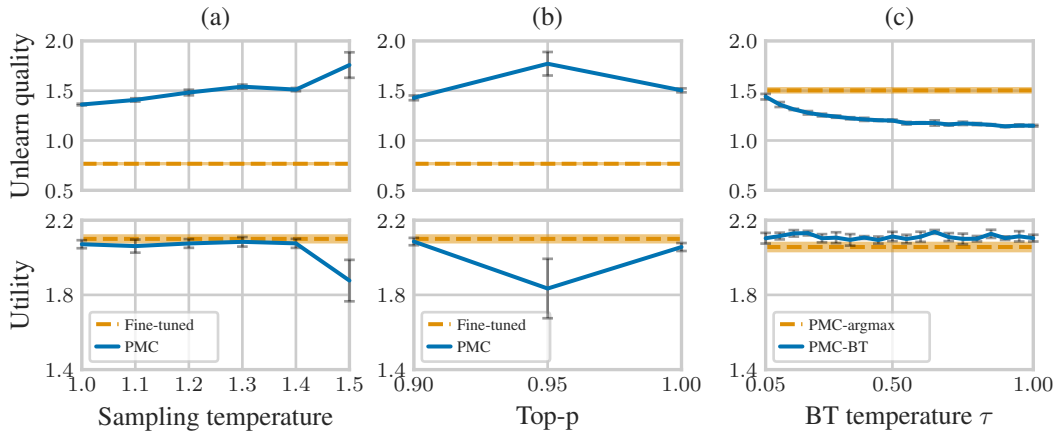

Figure 7: Ablation studies on: (a) temperature, (b) top-p sampling, and (c) Bradley-Terry approximation (shadow/bars show standard deviation across five runs).

A Additional experiments

We conduct the following additional ablation studies for Phi-1.5 to further investigate the properties of our method. See Appendix B for details on the experimental setup of all ablation studies.

Sampling temperature (Figure 7 (a)). We empirically observe that larger temperatures allow stronger unlearn quality, likely due to the higher probability to sample responses with lower similarity to the ground truth. For temperatures above 1.5, the process leads to a decrease in utility.

Top-p sampling (Figure 7 (b)). We observe that top-p sampling can similarly improve unlearn quality at the cost of model utility due to similar effects on the diversity of the sampled responses.

Bradley-Terry approximation. The general PMC formulation in Section 4 requires randomly selecting one of n responses according to the Bradley-Terry model. We implement an approximation by choosing the response with the highest score, i.e., $\hat{x} = \arg\max_i r(x_i)$. We resolve ambiguity by choosing whichever sample has been drawn first to provide an additional, implicit incentive to choose more likely samples. Our approximation is computationally efficient, and we empirically verify in Figure 7 (c) that it effectively corresponds to the limit $\tau \to 0$ for temperature τ. In detail, the BT-temperature τ does not affect model utility, however, it effectively improves unlearn quality, motivating the argmax approximation.

A.1 Extended utility experiments

We conducted additional experiments to test whether PMC introduces unexpected utility degradations on common benchmarks from the literature. Specifically, we compared the baseline Phi and Llama models with their PMC-unlearned counterparts on Arc-Challenge, Arc-Easy (Clark et al., 2018), and MMLU (Hendrycks et al., 2021). For PMC models, we report the mean and standard deviation over five random seeds. The results in Table 1 show that PMC has only minor impact on model utility.

	Arc-Challenge	**Arc-Easy**	**MMLU**
`Llama-3.2-3B-Instruct` (base)	0.4368	0.7382	0.6041
`Llama-3.2-3B-Instruct` (PMC)	0.4341 $\pm$ 0.0061	0.7230 $\pm$ 0.0058	0.5924 $\pm$ 0.0040
`Phi-1.5` (base)	0.4462	0.7622	0.4174
`Phi-1.5` (PMC)	0.4283 $\pm$ 0.0057	0.6891 $\pm$ 0.0053	0.4063 $\pm$ 0.0031

Table 1: Model utility comparison between baseline models and models fine-tuned with the proposed PMC method. Mean and standard deviation over 5 random seeds are shown for PMC.

B EXPERIMENTAL SETUP

We conduct all experiments on NVIDIA A100 GPUs (40GB) and NVIDIA H100 GPUs (80GB).

Datasets. We use the TOFU Q&A dataset (Maini et al., 2024) for finetuning and unlearning. The dataset consists of 4,000 question-answer pairs about generated autobiographies of 200 different, fictitious authors. We use the "forget10" split of the dataset, since it is the most challenging split of the dataset. The split uses 400 samples for the forget set and the remaining samples for the retain set. To facilitate model evaluation we approximate retain performance using a (random but fixed) subset of 400 retain samples.

B.1 FINETUNING DETAILS

We fine-tune two pretrained LLMs, Phi-1.5 (Li et al., 2023) and Llama-3.2-3B-Instruct (Grattafiori et al., 2024). For fine-tuning we generally follow the experimental setup as described in (Maini et al., 2024). We fine-tune both models on the full TOFU Q&A dataset (Maini et al., 2024).

Finetuning hyperparameters. We fine-tune both models for 5 epochs using the AdamW optimizer (Loshchilov & Hutter, 2019) together with ZeRO-3 (Rajbhandari et al., 2020). For fine-tuning Phi-1.5 we use a batch size of 16 and gradient accumulation steps of 2, which results in an effective batch size of 32. For Llama-3.2-3B-Instruct we use a batch size of 8 and gradient accumulation steps of 2, which results in an effective batch size of 16. We use a learning rate of $2e^{-5}$ for Phi-1.5 and $1e^{-5}$ for Llama-3.2-3B-Instruct. We also apply weight decay of 0.01 for both models. For Llama-3.2-3B-Instruct we additionally deploy gradient checkpointing (Chen et al., 2016) and disable flash attention. We summarize hyperparameters in Table 2 and results in Table 3.

Table 2: Finetuning hyperparameters for Phi-1.5 and Llama-3.2-3B-Instruct.

Finetuning Hyperparameter	Phi-1.5	Llama-3.2-3B-Instruct
Batch size	16	8
Gradient accumulation steps	2	2
Learning rate	2e-5	1e-5
Number of epochs	5	5
Weight decay	0.01	0.01
Gradient checkpointing	False	True

Table 3: ROUGE-L scores as well as unlearn quality (UQ) and utility (as defined in Section 5) for the pretrained models (before finetuning) and the models after finetuning on the TOFU 90/10 split.

Model	**Full**	**World-facts**	**Real-authors**	**Forget**	**Paraph. forget**	**Retain**	**UQ**	**Utility**
`Phi-1.5`	0.45	0.82	0.66	0.45	0.39	0.45	1.16	1.93
`Phi-1.5` (FT)	0.93 ± 0.00	0.75 ± 0.02	0.44 ± 0.01	0.92 ± 0.00	0.31 ± 0.00	0.91 ± 0.01	0.77 ± 0.00	2.10 ± 0.03
`Llama-3.2-3B-I.`	0.26	0.92	0.96	0.27	0.24	0.26	1.49	2.14
`Llama-3.2-3B-I.` (FT)	0.96 ± 0.00	0.90 ± 0.01	0.86 ± 0.02	0.95 ± 0.00	0.34 ± 0.00	0.96 ± 0.00	0.71 ± 0.00	2.72 ± 0.02

B.2 UNLEARNING DETAILS

For the unlearning experiments we use the same hyperparameters for finetuning, except when otherwise stated in the grid search. For fair comparison between methods, we run 100 experiments for each method. We repeat each experiment 5 times using the same fixed random seeds for all methods and report mean across the runs. That is we run 500 experiments for each method. We summarize the hyperparameters used for the grid search in Table 4. Note that we introduce λ as a trade-off between retain and forget loss for all methods, even if their original formulation does not include it.

Runtime. We report the runtime of PMC-unlearning for the two different models. On an NVIDIA H100 GPU, `Phi-1.5` completes within 40 ± 2 minutes, whereas `Llama-3.2-3B-Instruct` converges faster with a runtime of 30 ± 1 minutes (both averaged over 5 runs). All other unlearning methods have slightly shorter runtimes and complete within 20-30 minutes. We consider the runtime of PMC to be reasonable for practical applications, especially when compared to retraining LLMs from scratch. We also believe that the runtime of PMC can be further improved by future research e.g. on more efficient sampling strategies.

Table 4: Gridsearch details for all unlearning methods. For a fair comparison, we run 500 experiments for each method: 100 different configurations each repeated for 5 different seeds. LR: Learning rate.

Parameter	**GA**
Seed	range(0,5)
LR	linspace(1e-5, 1e-4, 10)
Epochs	linspace(2, 20, 10)

Parameter	**GD**
Seed	range(0,5)
LR	{1e-5, 2e-5}
Epochs	{3, 5, 10, 15, 20}
λ	linspace(0.5, 1.5, 10)

Parameter	**IDK**
Seed	range(0,5)
LR	{1e-5, 2e-5}
Epochs	{3, 5, 10, 15, 20}
λ	linspace(0.5, 1.5, 10)

Parameter	**NPO**
Seed	range(0,5)
LR	1e-05
Epochs	10
λ	linspace(0.5, 1.5, 10)
β	linspace(0.05, 0.2, 10)

Parameter	**SimNPO**
Seed	range(0,5)
LR	1e-05
Epochs	10
λ	linspace(0.05, 0.25, 4)
β	linspace(2.5, 5.5, 5)
γ	linspace(0.0, 2.0, 5)

Parameter	**PMC** (Phi-1.5)
Seed	range(0,5)
LR	1e-05
Epochs	{10, 15}
λ	linspace(0.5, 1.5, 5)
#Samples	{1, 5, 10, 15, 20}
Temperature	{1.25, 1.5}
Top-p	0.95

Parameter	**PMC** (Llama-3.2-3B-Instruct)
Seed	range(0,5)
LR	1e-05
Epochs	{15, 20}
λ	{0.5, 0.75, 1, 2, 3}
#Samples	{10, 15}
Temperature	{0.8, 0.9, 1, 1.25, 1.5}
Top-p	0.95

B.3 MPC PROMPT TEMPLATE

We created the MPC dataset from the TOFU Q&A dataset by prompting ChatGPT to do this specific task. We selected a subset of 84 questions based on their suitability to be converted to a multiple choice format. Suitability was evaluated using ChatGPT with the following template:

> Answer with either 'Yes' if the following is a factual question e.g., it can be answered with a few words, such as names, dates, orientation, etc., or 'No' if it requires longer explanations. Do not output anything beyond 'Yes', or 'No'.
>
> **QUESTION:** {question}
> **ANSWER:** {answer}

The prompt template used to convert the dataset to MPC is shown in the following:

> Convert the following question and answer into a multiple choice question with 4 possible answers. For each option remain close to the original sentence structure. Here is an example of an original question and answer:
>
> **QUESTION:** What is the full name of the author born in Taipei, Taiwan on 05/11/1991 who writes in the genre of leadership?
> **ANSWER:** The author's full name is Hsiao Yun-Hwa.
>
> What should be generated in this case:
>
> **MPC ANSWER:**
> A) The author's full name is Hsiao Yun-Hwa.
> B) The author's full name is Ming-Chi Lee
> C) The author's full name is Wei-Li Chen
> D) The author's full name is Yu-Ting Huang
>
> **CORRECT ANSWER:** A Do it for the following pair:
>
> **QUESTION:** {question}
> **ANSWER:** {answer}
>
> **MPC ANSWER:**

B.4 Details on ablation studies

Table 5: Overview of hyperparameters used for the ablation studies in Section 5 and Appendix A. For each setting, we repeat each experiments for 5 different seeds and report mean and standard deviation. All ablation studies are conducted on the Phi-1.5 model.

Number of epochs

Parameter	NPO
Seed	range(0,5)
LR	1e-05
Epochs	range(1,21)
λ	1.5
β	0.05

Number of epochs

Parameter	SimNPO
Seed	range(0,5)
LR	1e-05
Epochs	range(1, 21)
λ	0.25
β	4
γ	0

Number of epochs

Parameter	PMC
Seed	range(0,5)
LR	1e-05
Epochs	range(1, 21)
λ	1.5
#Samples	5
Temperature	1.25
Top-p	0.95

Number of samples

Parameter	PMC
Seed	range(0,5)
LR	1e-05
Epochs	15
λ	1.5
#Samples	range(1,21)
Temperature	1.25
Top-p	0.95

λ

Parameter	PMC
Seed	range(0,5)
LR	1e-05
Epochs	15
λ	range(0.5, 1.55, 0.1)
#Samples	5
Temperature	1.25
Top-p	0.95

Sampling temperature

Parameter	PMC
Seed	range(0,5)
LR	1e-05
Epochs	15
λ	1.5
#Samples	5
Temperature	range(1.0, 1.55, 0.1)
Top-p	0.95

Top-p sampling

Parameter	PMC
Seed	range(0,5)
LR	1e-05
Epochs	15
λ	1.5
#Samples	5
Temperature	1.25
Top-p	{0.9, 0.95, 1}

BT temperature τ

Parameter	PMC
Seed	range(0,5)
LR	1e-05
Epochs	15
λ	1.5
#Samples	5
Temperature	1.25
Top-p	0.95
τ	linspace(0.05, 1, 20)

B.5 Hyperparameter details on sampling and prefilling experiment

For the sampling and prefilling experiment (Subsection 5.3) we train Llama-3.2-3B-Instruct models for three different unlearning techniques (PMC, IDK, NPO) with the following hyperparameters: For all methods we use a learning rate of $1e^{-5}$ and 20 epochs. For PMC and IDK we choose $\lambda = 1.25$. For PMC we use 20 samples, a temperature of 0.9, and top-p of 0.95 during PMC sampling. For NPO we use $\lambda = 1.5$ and $\beta = 0.05$. We also repeated the experiment with SimNPO, but the results were very similar to NPO. After unlearning, we sample 100 responses per question from each model using a temperature of 0.9 and top-p of 0.95. We then compute the ROUGE-L score between each sampled response and the ground truth answer. We report the worst-case ROUGE-L score average over all questions in the forget set and report it in Figure 6. The worst-case ROUGE-L score for a question is defined as the maximal ROUGE-L score across all 100 sampled responses for that question.

C WARM-UP: ITERATIVE UNLEARNING WITH CATEGORICAL DISTRIBUTIONS (SECTION 4)

Definition C.1 (Categorical distribution)**.** A categorical distribution is a probability distribution over K different possible outcomes $\{0, \dots, K-1\}$ and parametrized by a vector $\pi = (p_0, \dots, p_{K-1})$ of probabilities for each category, where $p_k \geq 0$ and $\sum_{k=0}^{K-1} p_k = 1$. The probability mass function is given by $\Pr[X = k] = p_k$.

Definition C.2 (Model collapse)**.** A random variable X is said to have a *collapsed* distribution if its variance is zero, i.e. $\mathrm{Var}[X] = 0$.

Learning a categorical distribution. Consider a random variable X equipped with a categorical distribution over K categories. We can learn the parameters π of the distribution of X from n realizations $x = (x_1, \dots, x_n)$ using maximum likelihood estimation (MLE). The likelihood function is given by

$$L(x;\pi) \triangleq \prod_{i=1}^{n} \Pr[X = x_i] = \prod_{k=0}^{K-1} p_k^{n_k} = \left(1 - \sum_{k=0}^{K-2} p_k\right)^{n_{K-1}} \prod_{k=0}^{K-2} p_k^{n_k}$$

where $n_k = \sum_{i=1}^{n} \mathbb{1}[x_i = k]$ is the number of times category k was observed, and in the last equation we rewrote p_{K-1} by making use of the fact that $\sum_{k=0}^{K-1} p_k = 1$. We maximize the log-likelihood function as follows:

$$\frac{\partial \log L(x;\pi)}{\partial p_k} = \frac{n_k}{p_k} - \frac{n_{K-1}}{1 - \sum_{i=0}^{K-2} p_i} \stackrel{!}{=} 0$$

$$\Leftrightarrow \quad p_k = \frac{n_k}{n_{K-1}} \left(1 - \sum_{i=0}^{K-2} p_i\right) \quad \Leftrightarrow \quad p_k + \frac{n_k}{n_{K-1}} \sum_{i=0}^{K-2} p_i = \frac{n_k}{n_{K-1}},$$

which is a linear system of $K-1$ equations. We can briefly verify that the solution to this linear system is given by $\hat{p}_k = \frac{n_k}{n}$:

$$\frac{n_k}{n} + \frac{n_k}{n_{K-1}} \sum_{i=0}^{K-2} \frac{n_i}{n} = \frac{n_k}{n_{K-1}} \left(\frac{n_{K-1}}{n} + \frac{\sum_{i=0}^{K-2} n_i}{n}\right) = \frac{n_k}{n_{K-1}}.$$

That is the MLE for the probability p_k of category k is the fraction $\frac{n_k}{n}$ of observing category k among all n samples.

Iterative relearning categorical distributions. Given an arbitrary categorical distribution with parameters π_0, we analyze iterative relearning of a categorical distribution on its own generated data. First we draw n samples $x = (x_1, \dots, x_n)$ i.i.d. from the distribution given by π_0. We then relearn the parameters π_1 from the dataset x via maximum likelihood estimation. Repeating this process will lead to convergence as we show in the following:

Proposition C.3. *Iteratively relearning of a categorical distribution π_t on its own generated data yields model collapse independent of the initial distribution.*

Intuitively, given finite samples, the iterative relearning process describes an absorbing Markov chain, which is known to converge to an absorbing state (Shumailov et al., 2023).

Full proof. For the sake of exposition we first consider the case of a categorical distribution with $K = 2$ categories, i.e. a Bernoulli distribution with a single success parameter p. Without loss of generality we further assume that the initial success probability is already a multiple of $\frac{1}{n}$ (otherwise just relearn once and then follow the proof).

The main idea of this proof is to model the stochastic process of relearning on self-generated data as a discrete-time discrete-state-space Markov chain. Specifically, during iterative relearning, the maximum likelihood estimate (average number of successes) itself becomes a random variable that defines the parameter for the distribution of the next iteration. We denote the number of successes in the $(t+1)$-th iteration as $Y_{t+1} = \sum_{i=1}^{n} X_t^{(i)}$, where $X_t^{(i)} \sim \mathrm{Ber}\left(\frac{Y_t}{n}\right)$ are i.i.d. Bernoulli random variables with success probability $\frac{Y_t}{n}$.

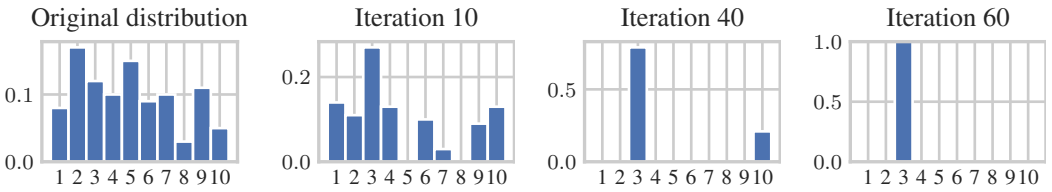

Figure 8: Model collapse for iterative retraining with categorical distributions. After 60 iterations, the distribution collapsed to a zero-variance distribution, i.e. a single category. Compare to Figure 2.

Note that there are only $n+1$ possible Bernoulli distributions because we estimate the success probability with a discrete value. Thus the stochastic process of iterative relearning can be described as a Markov chain with state space $\{0, 1, \ldots, n\}$ corresponding to the $n+1$ possible Bernoulli distributions. We further describe the stochastic process using a $(n+1) \times (n+1)$ transition matrix $P_n = (p_{ij})$ of the probabilities to transition from one distribution to another:

$$p_{ij} \triangleq \Pr\left[Y_{t+1} = j \mid Y_t = i\right] = \text{Binom}\left(j;\ n, i/n\right).$$

In other words, the rows of the transition matrix corresponds to the PMF of the Binomial distribution with n samples and success probability i/n, where i corresponds to the number of successes of the previous iteration.

As an example, we show transition matrices for $n = 1, 2, 3$ samples:

$$P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 1/4 & 1/2 & 1/4 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.29629630 & 0.44444444 & 0.22222222 & 0.03703704 \\ 0.03703704 & 0.22222222 & 0.44444444 & 0.29629630 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Notably, the described Markov chain is a so-called absorbing Markov chain: First, it contains absorbing states (0 and n) corresponding to Bernoulli distributions with success probability zero or one – once a random walker reaches one of the absorbing states, the walker cannot leave it anymore. Second, it is possible to go from any transient (non-absorbing) state to an absorbing state in a finite number of steps. Thus a random walker is guaranteed to eventually reach an absorbing state, independent of the initial success probability.

Consequently, iterative relearning will result w.p.1 in a distribution with success probability zero or one. Since the variance of a Bernoulli distribution is $p(1-p)$, the variance of the final distribution is zero, i.e., the distribution collapsed.

For the general case of a categorical distribution with K categories, the proof follows analogously by considering the Markov chain with states corresponding to the possible $\binom{n+K-1}{K-1}$ categorical distributions $\mathbf{p}$. In this case, the rows correspond to the PMF of a Multinomial distribution: $p_{ij} = \text{Multinom}\left(n\mathbf{p}[j];\ n, \mathbf{p}[i]\right)$, where $\mathbf{p}[i]$ denotes the i-th categorical distribution in the state space. The absorbing states correspond to the K distributions with $p_k = 1$ for one k and $p_i = 0$ for all other i, which again have zero variance, i.e. are collapsed distributions. □

Proposition C.3 is a special case of the argument of Shumailov et al. (2023) that iterative relearning with discrete distributions describes an absorbing Markov chain, which is known to converge to absorbing states with probability 1. Our proof explicitly constructs the underlying absorbing Markov chain for categorical distributions.

Expected number of steps until model collapse. Interestingly, with a single sample the transition matrix is the identity matrix and the distribution collapses immediately. In general, more samples means slower collapse. Specifically, the expected steps until model collapse corresponds to the expected steps to reach an absorbing state and can be computed by the fundamental matrix $\sum_{t=0}^{\infty} Q^t$, where Q is the submatrix of the transition matrix P corresponding to the transient states.

Notably, the submatrix Q is a strictly substochastic matrix, i.e. the sum of the entries in each row is strictly less than one (since it does not contain the non-zero probability of transitioning to absorbing states). We can bound the eigenvalues of Q using the Gershgorin circle theorem (Geršgorin, 1931), which states that every eigenvalue of a square matrix M lies within a closed disk centered at M_{ii} with radius R_i, where M_{ii} is the diagonal element of M and R_i is the sum of the absolute values of the off-diagonal elements of row i, $R_i = \sum_{j \neq i} |M_{ij}|$. In our case, since Q is substochastic, the absolute eigenvalues of Q are strictly less than one. This allows us to apply the geometric series of matrices and compute the fundamental matrix as $\sum_{t=0}^{\infty} Q^t = (I - Q)^{-1}$. The expected number of steps until model collapse can be computed by solving the linear system $(I - Q)\mathbf{t} = \mathbf{1}$. Overall, starting in transient state i, the expected number of steps until model collapse is given by $\mathbf{t_i}$. In Figure 9 we show that the expected steps $\mathbf{t}_i$ grows linearly with the number of samples n.

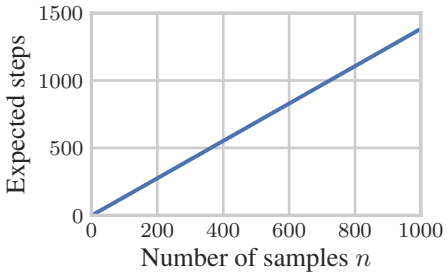

Figure 9: Expected number of steps until model collapse for Bernoulli distributions.

From model collapse to machine unlearning for categorical distributions.

Lemma 1: *For any categorical distribution π_0, iteratively relearning π_t on target data $\mathcal{D}_C$ augmented with data generated from its own distribution $\{x_i \mid x_i \sim \pi_t\}_{i=1}^n$ causes information loss for all other (non-target) categories i: $\pi_t(i) \xrightarrow{t\to\infty} 0$.*

Proof. Because of the fixed retain set, probabilities for retain categories remain non-zero, while probabilities for all other categories can become zero if no samples from these categories are generated during the iterative relearning process. Once the probability of a category becomes zero, it cannot be recovered anymore, since the iterative relearning process only generates samples from the current distribution π_t and will not generate samples from categories that have zero probability. This process can be described once again using an absorbing Markov chain, where the absorbing states correspond to the distributions with zero probability for all categories except the retain categories. □

Beyond categorical distributions. We empirically demonstrate partial collapse in finite samples for distributions described by Gaussian mixture models (GMMs) for 1- and 2-dimensional data. Specifically, we sample two datasets from two isotropic Gaussians, one retain and one forget set. We then fit a GMM with two Gaussians on the joint dataset to obtain a starting distribution p_0. We then iteratively relearn the GMMs either (1) on datapoints sampled from the model's own distribution only, or (2) on retain data augmented with datapoints sampled from the model's own distribution. Figure 10 and Figure 11 show that iterative relearning on self-generated data leads to information loss – either the distribution collapses to zero variance or the variance diverges. In contrast, Figure 12 and Figure 13 show that iterative relearning on retain points augmented with self-generated data leads to partial collapse, i.e. the probability mass of the forget distribution is redistributed to the retain distribution. This process stabilizes and does not collapse. This is consistent with the observation for categorical distributions in Figure 2 (collapse) and Figure 8 (partial collapse/unlearning).

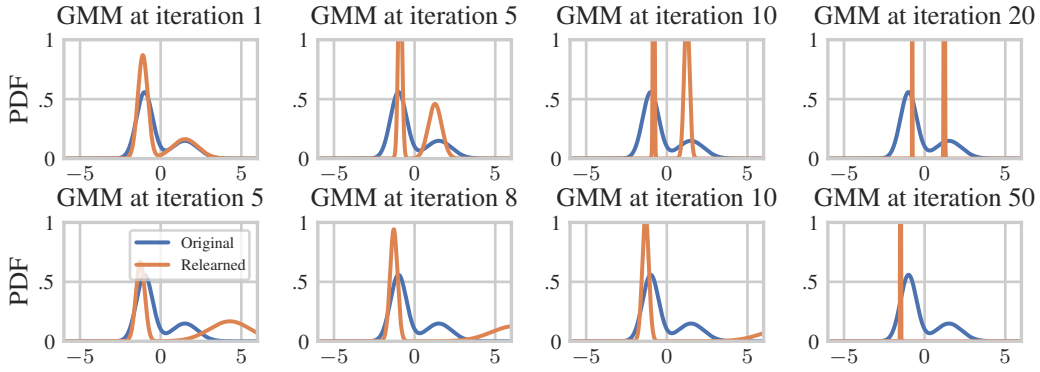

Figure 10: Model collapse during iterative relearning of 1D-GMMs without retain set. Variance of each individual Gaussian either converges to 0 (top row) or diverges to ∞ (bottom row) in finite steps.

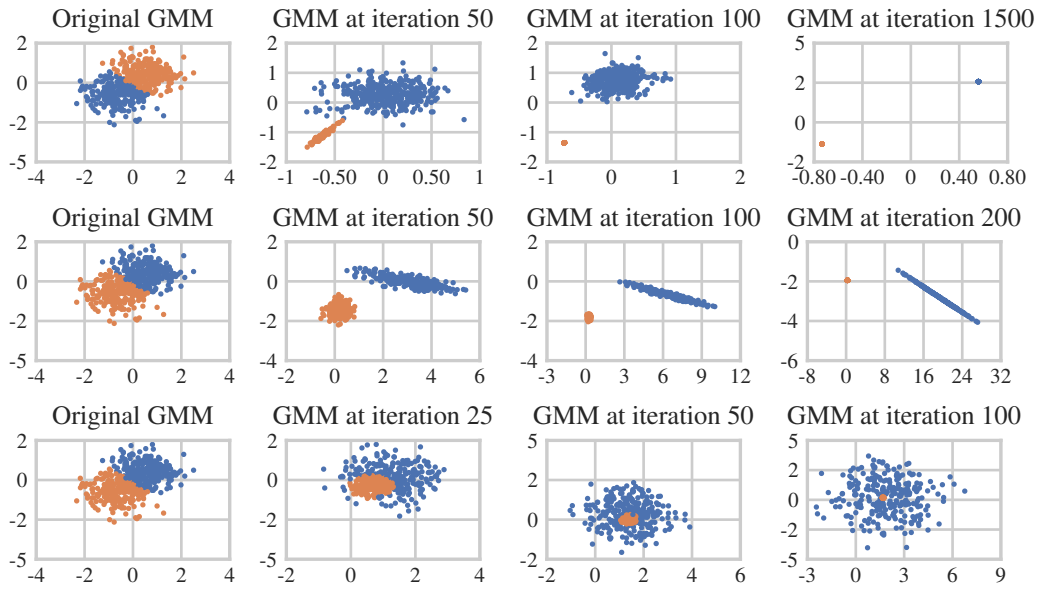

Figure 11: Model collapse during iterative relearning of 2D-GMMs without retain set. Variance of each individual Gaussian either vanishes or diverges (in finite steps).

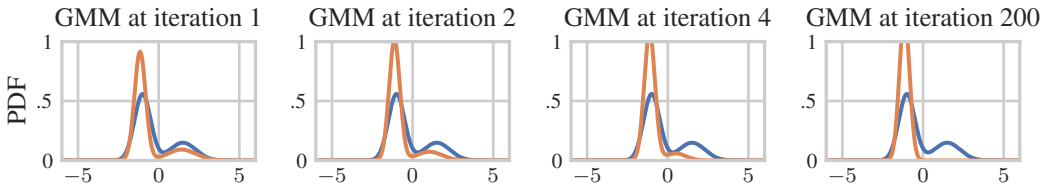

Figure 12: Partial model collapse unlearning for 1D-GMMs: When augmenting retain data with self-generated data, the probability mass of the forget distribution is redistributed to the retain distribution. The iterative relearning process stabilizes and does not collapse.

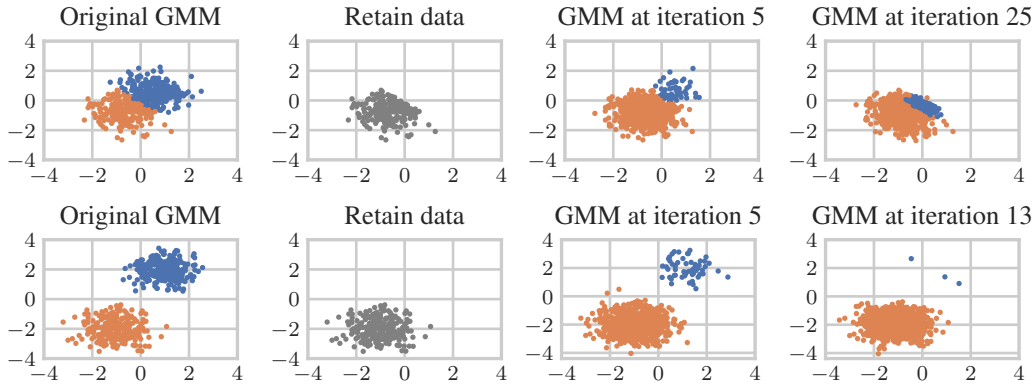

Figure 13: Partial model collapse unlearning for 2D-GMMs: When augmenting retain data with self-generated data, the probability mass of the forget distribution is redistributed to the retain distribution. The iterative relearning process stabilizes and does not collapse. Note that singularities in the EM-algorithm may occur during this iterative process (bottom row).

D MACHINE UNLEARNING VIA RELEARNING ON SELF-GENERATED DATA

The approach we describe in Section 4 is specific for Q&A tasks, but we can generalize it into unlearning for arbitrary tasks as well: Let p_0 denote the PDF (PMF) of any distribution over a set $\mathcal{X} \subseteq \mathbb{R}^d$. Starting from an initial distribution, the objective is to obtain a model that fits a target distribution p_r, while erasing the influence of a forget distribution p_f. Given a target distribution p_r over $\mathcal{X}$ that we do not want to unlearn, we propose machine unlearning via iterative relearning as:

Partial Model Collapse Machine Unlearning

$$p_{t+1} = \arg\min_{p \in \mathcal{P}(\mathcal{X})} \frac{\alpha}{1+\alpha} \mathbb{E}_{x \sim p_r}[-\log p(x)] + \frac{1}{1+\alpha} \mathbb{E}_{x \sim p_t}[-\log p(x)] \tag{3}$$

where $\mathcal{P}$ is the set of densities over $\mathcal{X}$, and $\alpha \in [0, \infty)$. Intuitively, Equation 3 describes an iterative process where the next distribution minimizes the convex combination of the expected negative log-likelihood (NLL) under a retain distribution and the expected NLL under the current distribution p_t. Notably, this iterative process converges to the retain distribution (Proof in Appendix D):

Theorem 2: *Assuming no statistical approximation errors, p_t converges exponentially with rate $\frac{1}{1+\alpha}$ to the target distribution p_r for any initial distribution p_0, $\lim_{t\to\infty} p_t(x) = p_r(x)$.*

Here, larger α yields faster convergence to the retain distribution p_r. Notably, we do not require any unlearning target, i.e. this method is independent of any forget distribution. In particular, Theorem 2 implies that for any forget distribution p_f over $\mathcal{X}$ the KL-divergence between p_t and p_f converges to the KL-divergence between retain and forget distribution: $D_{\mathrm{KL}}(p_t||p_f) \xrightarrow{t\to\infty} D_{\mathrm{KL}}(p_r||p_f)$.

Proof. Due to assumption 1, we can express the PDF of the iterative relearning scheme as follows:

$$p_t(x) = \frac{\lambda}{1+\lambda} p_r(x) + \frac{1}{1+\lambda} p_{t-1}(x)$$

since the assumption ensures $q = \arg\max_{p \in \mathcal{P}} \mathbb{E}_{x\sim q}[\log p(x)]$. Note this is a recursion equation for which we can derive a closed-form:

$$\begin{aligned}
p_t(x) &= \frac{\lambda}{1+\lambda} p_r(x) + \frac{1}{1+\lambda} p_{t-1}(x) \\
&= \frac{\lambda}{1+\lambda} p_r(x) + \frac{1}{1+\lambda} \left(\frac{\lambda}{1+\lambda} p_r(x) + \frac{1}{1+\lambda} p_{t-2}(x) \right) \\
&\vdots \\
&\overset{(1)}{=} \frac{\lambda}{1+\lambda} \sum_{i=0}^{t-1} \left(\frac{1}{1+\lambda} \right)^i p_r(x) + \left(\frac{1}{1+\lambda} \right)^t p(x) \\
&\overset{(2)}{=} \frac{\lambda}{1+\lambda} \frac{1 - \left(\frac{1}{1+\lambda} \right)^t}{1 - \frac{1}{1+\lambda}} p_r(x) + \left(\frac{1}{1+\lambda} \right)^t p(x) \\
&\overset{(3)}{=} \left[1 - \left(\frac{1}{1+\lambda} \right)^t \right] p_r(x) + \left(\frac{1}{1+\lambda} \right)^t p(x)
\end{aligned}$$

where in (1) we insert the initial distribution $p_0(x) = p(x)$ after unrolling all t iterations, in (2) we use the geometric sum using $\lambda > 0$ and thus $\frac{1}{1+\lambda} \in (0, 1)$, and in (3) we just simplify the expression $\frac{1}{1-\frac{1}{1+\lambda}} = \frac{1+\lambda}{\lambda}$.

Thus we have derived a closed-form of $p_t(x)$:

$$p_t(x) = \left[1 - \left(\frac{1}{1+\lambda} \right)^t \right] p_r(x) + \left(\frac{1}{1+\lambda} \right)^t p(x)$$

Using this closed-form of $p_t(x)$ we directly obtain the convergence of $p_t(x)$ for $t \to \infty$:

$$p_\infty \triangleq \lim_{t\to\infty} p_t(x) = p_r(x)$$

since due to $\lambda > 0$ we have $\frac{1}{1+\lambda} \in (0,1)$ and thus $\left(\frac{1}{1+\lambda}\right)^t \xrightarrow{t\to\infty} 0$.

Consequently we further have:

$$D_{\mathrm{KL}}(p_\infty||p_r) = \mathbb{E}_{p_\infty}\left[\log \frac{p_\infty(x)}{p_r(x)}\right] = \mathbb{E}_{p_r}\left[\log \frac{p_r(x)}{p_r(x)}\right] = \mathbb{E}_{p_r}\left[\log 1\right] = 0$$

and

$$D_{\mathrm{KL}}(p_\infty||p_f) = \mathbb{E}_{p_\infty}\left[\log \frac{p_\infty(x)}{p_f(x)}\right] = \mathbb{E}_{p_r}\left[\log \frac{p_r(x)}{p_f(x)}\right] = D_{\mathrm{KL}}(p_r||p_f)$$

and specifically for mutually exclusive support of p_r and p_f we have:

$$D_{\mathrm{KL}}(p_\infty||p_f) = D_{\mathrm{KL}}(p_r||p_f) = \infty$$

□

Finally, we prove the theorem about the expected reward convergence and vanishing variance for the iterative relearning as described by Equation 2:

$$p_{t+1} = \arg\max_{p\in\mathcal{P}} \lambda\mathbb{E}_{(q,x)\sim p_r}[\log p(x|q)] + \mathbb{E}_{\substack{q\sim p_f \\ x_1,\ldots,x_n\sim p_t(x|q) \\ \hat{x}\sim\mathcal{BT}_\tau(x_1,\ldots,x_n)}}[\log p(\hat{x}|q)]$$

Theorem 1: *Let p_t be the distribution described by Equation 2. In the absence of statistical and function approximation errors, the expected reward converges to the maximum reward and its variance vanishes for any forget query $q \in supp(p_f)$:*

$$\mathbb{E}_{x\sim p_t(x|q)}\left[e^{r(x)}\right] \xrightarrow{t\to\infty} e^{r^*} \qquad \mathrm{Var}_{x\sim p_t(x|q)}\left[e^{r(x)}\right] \xrightarrow{t\to\infty} 0.$$

Proof. We consider the following iterative optimization problem (Equation 2):

$$p_{t+1} = \arg\max_{p\in\mathcal{P}} \lambda\mathbb{E}_{(q,x)\sim p_r}[\log p(x|q)] + \mathbb{E}_{\substack{q\sim p_f \\ x_1,\ldots,x_n\sim p_t(x|q) \\ \hat{x}\sim\mathcal{BT}(x_1,\ldots,x_n)}}[\log p(\hat{x}|q)]$$

Assuming no statistical approximation errors, we know that $\arg\max_{p\in\mathcal{P}} \mathbb{E}_{x\sim q}[\log p(x)] = q$. In the case of conditional distributions we have $\arg\max_{p\in\mathcal{P}} \mathbb{E}_{(q,x)\sim p_r}[\log p(x|q)] = p^*$ with $p^*(x|q) = p_r(x|q)$. Since we assume that the supports of p_r and p_f are disjoint, the optimization problem amounts to two independent problems and the density of the optimal distribution p^*_{t+1} matches each conditional distribution independently:

$$p^*_{t+1}(x|q) = \begin{cases} p_r(x|q) & \text{if } q \in supp(p_r) \\ \hat{p}_{t+1}(x|q) & \text{if } q \in supp(p_f) \end{cases}$$

where $\hat{p}(x|q)$ is the distribution that maximizes the second term in Equation 2 for $q \in supp(p_f)$:

$$\hat{p}_{t+1}(x|q) = \arg\max_{p\in\mathcal{P}} \mathbb{E}_{\substack{q\sim p_f \\ x_1,\ldots,x_n\sim\hat{p}_t(x|q) \\ \hat{x}\sim\mathcal{BT}(x_1,\ldots,x_n)}}[\log p(\hat{x}|q)]$$

Assuming again no statistical approximation errors, one can show that the density of the distribution $\hat{p}_{t+1}(x|q)$ assumes a closed-form (proof in (Ferbach et al., 2024) – proof of Lemma 2.1):

$$\hat{p}_{t+1}(x|q) = \hat{p}_t(x|q) \cdot H^n_{\hat{p}_t}(x|q)$$

with

$$H^n_{\hat{p}_t}(x|q) = \mathbb{E}_{x_1,\ldots,x_{n-1}\sim\hat{p}_t(x|q)}\left[\frac{ne^{r(x)}}{e^{r(x)} + \sum_{i=1}^{n-1} e^{r(x_i)}}\right].$$

Moreover, since we assume the reward is bounded, Assumption 2.1 in (Ferbach et al., 2024) holds and consequently the statement about reward convergence and vanishing variance follows directly from Lemma 2.2 in (Ferbach et al., 2024). □