GENERALIZED INFERENCE TIME UNLEARNING — EFFECTIVE FOR A FRACTION OF THE COST

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) can memorize and regurgitate sensitive training data, creating significant privacy and safety risks. While existing unlearning aim to address these risks, current methods are often computationally prohibitive and/or significantly degrade model utility. We introduce a framework for Inference-Time Unlearning, a new paradigm that steers an LLM's output at inference time using small secondary models, without altering the base model's weights. Through extensive experiments with LLMs we demonstrate that our method is highly effective at removing targeted verbatim and semantic knowledge, is orders of magnitude more computationally efficient than traditional approaches, and fully preserves the base model's general capabilities. We then explore efficacy in unlearning visual semantics in generative image models and find similar evidence of effectiveness. Finally, we introduce a new benchmark focused on unlearning time-dependent information. Collectively, the framework offers a practical, scalable, and low-cost solution for selective forgetting, enabling more responsible and adaptable model deployment. All code to reproduce this work is available at https://anonymous.4open.science/r/inference-time-unlearning-iclr2026/

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable capabilities, achieving state-of-the-art performance on a diverse array of natural language tasks and becoming integral to a wide range of applications (Brown et al., 2020; Touvron et al., 2023; DeepSeek-AI, 2024). However, the very scale that enables their powerful generalization also creates significant challenges (Weidinger et al., 2021). LLMs have been shown to memorize and regurgitate portions of their training data, including personally identifiable information (PII), proprietary text, and harmful content (Carlini et al., 2021). This behavior creates urgent privacy, safety, and copyright concerns (Henderson et al., 2023), conflicting with principles like the "right to be forgotten" mandated by regulations such as the GDPR (Voigt & dem Bussche, 2017).

The most straightforward solution to remove unwanted data from an LLM is to retrain it from scratch on a sanitized dataset (Bourtoule et al., 2021). Given that training a flagship model requires vast computational resources, this approach is economically and practically infeasible for frequent unlearning requests. For instance, training Meta's Llama 3 70B model consumed approximately 1.6 million GPU-hours, and other state-of-the-art models demand similarly massive-scale resources (Hoffmann et al., 2022; Grattafiori et al., 2024; DeepSeek-AI, 2025). Consequently, the field of machine unlearning has emerged to develop methods that can efficiently remove data's influence from a trained model (Nguyen et al., 2025). Prevailing techniques often rely on fine-tuning the full model, using methods like gradient ascent to maximize the likelihood of forgetting specific data or negative preference optimization to steer the model away from undesired outputs (Eldan & Russinovich, 2023; Jang et al., 2023). While less expensive than complete retraining, these methods still require costly gradient updates on the entire large model and can often lead to a degradation of the model's overall capabilities, a phenomenon known as catastrophic forgetting (Kirkpatrick et al., 2017).

In this work, we propose a new paradigm inspired by product of experts (Hinton, 1999) and speculative decoding (Leviathan et al., 2023): **Inference-Time Unlearning**. Our method, Divergence Decoding (DD), requires no modifications to the weights of the large base model. Instead, it guides

text generation at inference time by using a pair of much smaller, specialized models. One small model is fine-tuned on the data to be forgotten (the "forget set"), while another is tuned on a proxy for the data to be retained. By modifying the logits of the base model with the difference of the "retain" and "forget" models, our method steers the output distribution away from unwanted content while leaving general knowledge and model utility largely unaffected, effectively preventing the generation of targeted content.

Our paper makes three primary contributions to the literature on machine unlearning:

1. **Efficacy**: We demonstrate that Inference-Time Unlearning effectively removes both verbatim and semantic knowledge from a model. Our experiments on the MUSE (Shi et al., 2025) and TOFU (Maini et al., 2024) benchmarks show a significant reduction in the model's ability to recall targeted information from the forget set. Further, we apply our method to VQGAN image generation models (Esser et al., 2021) and find some evidence of unlearning visual semantics.

2. **Efficiency**: By restricting fine-tuning to small models (with orders of magnitude fewer parameters than the base LLM), our approach drastically reduces the computational cost of unlearning. For example, we find that even simple tri-gram based LMs are effective. This makes on-demand unlearning practical and scalable.

3. **Utility Preservation**: Our method maintains the model's performance on general knowledge and standard evaluation benchmarks. Because the base model's weights remain unchanged, the impact on its core capabilities is minimal, outperforming prior methods in preserving utility as the number of unlearning requests grows.

We show that our approach provides a practical, low-cost, and effective solution to the critical problem of selectively forgetting information in LLMs, paving the way for more responsible and adaptable deployment of these powerful models.

2 Related Literature

2.1 Alignment and Unlearning

Most unlearning methods are performed on the model's weights. Model providers use methods such as Supervised Safety Fine-tuning and RLHF to finetune their models to reduce the likelihood of generating certain content when aligning the models (Touvron et al., 2023; Achiam et al., 2024). For post-alignment methods, a variety of different variations of finetuning aim to remove knowledge from the model's weights while damaging its utility as little as possible. (Jang et al., 2023; Eldan & Russinovich, 2023; Zhang et al., 2024; Dong et al., 2024; Fan et al., 2024). While prior work has found that these methods *can* be effective, they are generally costly and almost always result in some utility loss.

Inference time approaches. Soft-prompting and in-context learning (Muresanu et al., 2024; Pawelczyk et al., 2024; Bhaila et al., 2025) aim to achieve unlearning by modifying the input to the model rather than the weights. However, these methods are still sensitive to changes in inputs and/or user behavior which may evolve over time, e.g., they can be jailbroken easily, since the knowledge is still inside the model. Further, the methods tend to be very niche/specialized use cases. Other approaches place classifiers or guardrails before and after the base model (Gao et al., 2025; Inan et al., 2023; Sharma et al., 2025).

Smaller models do not necessarily imply a loss of performance. Evidence from (Gunasekar et al., 2023; Bucher & Martini, 2024; Pecher et al., 2025) show that when finetuned for specialized tasks, small models can match or outperform the performance of general larger models. In addition, (Leviathan et al., 2023) proposed Speculative Decoding, demonstrating that smaller models can be used to accelerate inference in tandem with larger models. Our work extends this literature by introducing a method of unlearning which relies on small specialized models to guide a larger model away from undesirable output.

2.2 BENCHMARKS FOR UNLEARNING

A variety of unlearning benchmarks have been established in the literature. For example, Eldan & Russinovich (2023) introduce unlearning the Harry Potter books as a method for evaluation. Shi et al. (2025) release MUSE, which takes the Harry Potter challenge and adds a news dataset, focusing on six metrics related to both data owner and model provider metrics. TOFU (Maini et al., 2024) is similar to MUSE, focusing on Q&A, and it has a much smaller dataset sizes. WMDP (Li et al., 2024) introduces a benchmark for unlearning harmful content across different domains. Finally, Open Unlearning (Dorna et al., 2025) provides a comprehensive evaluation of many of the underlying metrics used for measuring unlearning and provides a harness for the testing and benchmarking of these methods. Collectively, the benchmarks above and others, comprehensively evaluate copyright, right to be forgotten, and toxic content generation.

While these benchmarks are focused on unlearning specific pieces of content, e.g., a news article for which the model provider does not own the copyright, there are other types of unlearning which are beneficial to the community. For example, LLMs are increasingly being used as surrogates for survey participants or generating forecasts. In such applications, it is common to want some sense of how well the model will perform out-of-sample. For this, we may need to unlearn specific time-dependent knowledge. Along these lines, in section 6 we extend this line of literature by introducing a benchmark for time-based unlearning.

3 METHOD

We begin by defining the problem, introducing our method, and finally connecting it to existing work. Let V denote a finite vocabulary of tokens. A token sequence of length T is denoted as $x = (x_1, x_2, ..., x_T)$ where each token $x_t \in V$. The prefix of a token sequence up to token $t-1$ is denoted $x_{<t} = (x_1, ..., x_{t-1})$. There are two data generating distributions D_A and D_B where the support of D_B is contained within D_A. Finally, $P(x_t|x_{<t})$ and $Q(x_t|x_{<t})$ denote the conditional token distributions under D_A and D_B, respectively.

We consider the situation where we wish to sample from Q but do not have access to it. Instead, only P is accessible. For example, P could be a large frontier model for which it is cost prohibitive to retrain a new model from scratch on D_B. Within the finance domain, Q could be a model as capable as P but trained up to a fixed knowledge cutoff so as to avoid look-ahead bias. Generally, our goal is to approximate sampling from Q using only P and samples drawn from D_A and D_B.

3.1 DIVERGENCE DECODING

Consider two small models $p(x_t|x_{<t})$ and $q(x_t|x_{<t})$ trained on samples from D_A and D_B, respectively. Denote the logits of a given model M as $l_M(x_{<t}) \in \mathbb{R}^{|V|}$. Divergence Decoding (DD) approximates sampling from Q by adjusting the logits of P according to the divergence between q and p. Empirically, we consider two adjustments. The first is a linear combination of the logits,

$$\hat{l}_Q^{LC}(x_{<t}) = l_P(x_{<t}) + \alpha \cdot [l_q(x_{<t}) - l_p(x_{<t})], \tag{1}$$

while the second adjustment is rank based,

$$\hat{l}_Q^{R}(x_{<t}) = l_P(x_{<t}) - \mathbb{1}_{rank(l_p(x_{<t}) - l_q(x_{<t})) \leq k} \cdot \infty. \tag{2}$$

In the case of the linear adjustment, if the difference between Q and P is indeed linear in logit space, then there exists some value of α, p, and q which enables Q to be perfectly recovered. If the difference is not linear however, then this is not true. For this reason, we also explore the rank based approach, which prevents generating the top-k most divergent tokens between p and q.

Samples can then be drawn via typical methods (e.g., Fan et al., 2018; Holtzman et al., 2020) from the approximation,

$$\widehat{Q}(x_t|x_{<t}) = \text{softmax}(\hat{l}_Q(x_{<t})). \tag{3}$$

While the adjustments in Eq. 1 and 2 require additional forward passes for p and q, we show in Section 4 that strong performance on certain tasks can be achieved even when p and q are trigram models—which add negligible computational overhead.

3.2 THEORETICAL MOTIVATION

While simple to implement and fast at inference time, our method is theoretically motivated by the Product of Experts (Hinton, 1999) and Importance Sampling (Hammersley & Handscomb, 1965) literature. In Appendix A.1, we show that the approximation $\widehat{Q}$ can be formulated as a Product of Experts model,

$$\widehat{Q}(x_t|x_{<t}) \propto \underbrace{P(x_t|x_{<t})}_{\text{Base Expert}} \cdot \underbrace{\left[\frac{q(x_t|x_{<t})}{p(x_t|x_{<t})}\right]^{\alpha}}_{\text{Domain Expert}} \tag{4}$$

where $\widehat{Q}$ is the product of a "Base Expert" P responsible for providing foundational knowledge and a "Domain Expert" comprised of the ratio of q to p. Intuitively, the role of the domain expert can be summarized by three cases:

1. $q \approx p$: Tokens are similarly likely under both D_A and D_B and the domain expert ratio is close to 1 effectively leaving the probabilities from the base model P unchanged
2. $q \gg p$: Tokens are much ***more*** likely under D_B than D_A, and the domain expert "upvotes" such tokens by ***increasing*** the probability assigned to them
3. $q \ll p$: Tokens are much ***less*** likely under D_B than D_A, and the domain expert "downvotes" such tokens by ***decreasing*** the probability assigned to them

Finally, DD can also be linked to importance sampling in Monte Carlo analysis whereby the expectation of some function $f(x)$ under a target distribution D_{target} is estimated using samples drawn from a proposal $D_{proposal}$. Formally,

$$\mathbb{E}_{x\sim D_{target}}[f(x)] = \mathbb{E}_{x\sim D_{proposal}}\left[f(x)\frac{D_{target}(x)}{D_{proposal}(x)}\right], \tag{5}$$

where the importance weight $w(x) = \frac{D_{target}(x)}{D_{proposal}(x)}$ adjusts the expectation taken over $D_{proposal}$ for differences between the proposal and target distributions. Analogously, divergence decoding uses the ratio of q to p to adjust for differences between the inaccessible model Q and accessible one P.

4 BENCHMARKS

Our primary unlearning experiments were conducted using the Open Unlearning framework (Dorna et al., 2025; Maini et al., 2024; Shi et al., 2024) on a cluster with two NVIDIA H100 GPUs. We adopt the MUSE vocabulary: the **Target** model refers to the model subject to unlearning, while **Retrain** denotes the best—but most costly—of retraining from scratch, which we aim to approximate.

We fine-tune one model on the retain set and one on the forget set. To reduce excessive divergence between p and q, the forget model may also include retain data if the retain set is much larger. For MUSE, specifically the news dataset, the retain set is about twice the size of the forget set, so training the forget model on only the forget set ends up outperforming. However, in TOFU's '90' benchmark, the retain set is approximately nine times larger than the forget set, in this case, we find that performance is improved by training on a mix of both sets of data.

We interpret privacy benchmarks as a measure of over- or under-unlearning. This interpretation is particularly relevant in settings such as backtesting, forecasting, and prediction, where overunlearning even without utility loss would skew the results. One caveat which is specific to the rank-based divergence decoding setup is that a naive implementation, i.e., one where logits for targeted tokens are set to $-\infty$, would lead to degenerate privacy scores since all the unlearned material will have logits of $-\infty$. As we still want to evaluate over- or under-unlearning for rank-based DD, for we instead replace the k most divergent logits with the kth largest logit in the original distribution. Of course, if an attacker has access to the logits, this substitution would be no better than the approach of setting masked logits to $-\infty$ since it is very unlikely that many tokens would have identical logit

values. As we discuss in section 7, a limitation of the *rank-based* method that it is not designed to be applied where the logits are publicly available and where the content being unlearned must be protected.

4.1 MUSE

Our first experiment leverages the news dataset from the MUSE benchmark (Shi et al., 2023; 2025). For the specialized p and q models, we finetune *princeton-nlp/Sheared-LLaMA-1.3B* (Xia et al., 2023), which shares a tokenizer with the official MUSE benchmark models (Shi et al., 2025). Our method exceeds—or achieves parity with—the unlearning frontier while preserving downstream utility (Figure 1). This performance is especially notable given the significantly reduced computational cost of our method relative to prior work. Detailed analysis of the setup, **hyper-parameter choices,** and the **scaling and sustainability benchmarks** are provided in Appendix B.1.

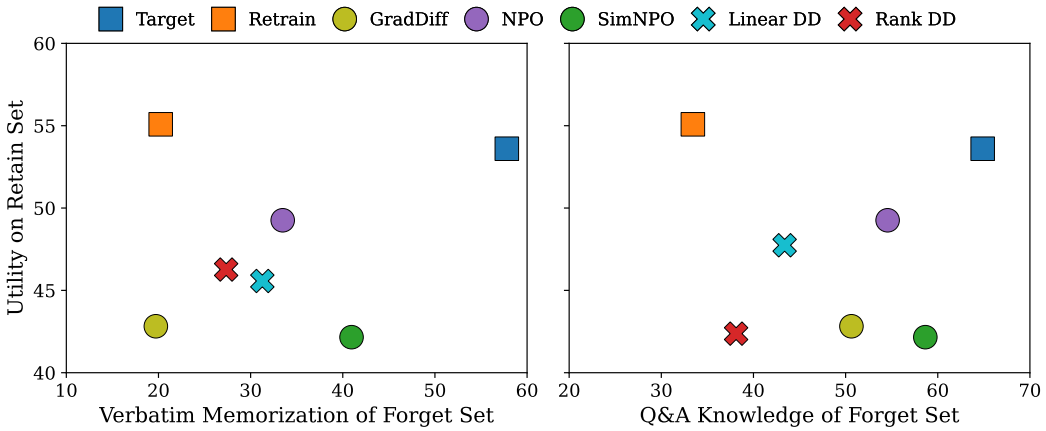


Figure 1: MUSE Results. Closer to retrain is better.

4.1.1 P AND Q MODEL SIZE

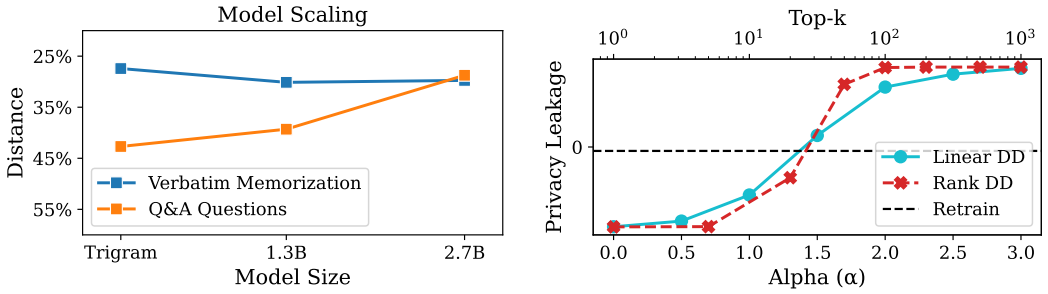
Given that applying our method using the 1.3B models for p and q is effective, a natural question is how sensitive this performance is to model size. We investigate this using *princeton-nlp/Sheared-LLaMA-2.7B* and trigram LMs based on *Stupid Backoff* (Brants et al., 2007). We select the most optimal configuration of every model size, based on the minimum euclidean distance to Retrain, and rescale the metric such that Target is 100%. The Trigram models outperform on the Verbatim Memorization and perform slightly worse than the LLMs on Q&A. Upon further inspection of the Q&A questions where the Trigram models perform well, we find that this is largely due to questions which are more similar to the underlying training data. Thus, we conclude that the Trigram models are likely most useful for unlearning verbatim content.

4.1.2 SUSTAINABILITY AND SCALING

Finally, prior work has found that many unlearning methods exhibit poor scalability—the unlearning of very large amounts of content—and sustainability—sequential requests to unlearn additional content. We explore the efficacy of our method along these dimensions using the MUSE scaling and sustainability benchmarks to ensure that performance does not degrade. To extend the benchmark, we additionally measure performance on the original forget set (Q&A), ensuring that improved generalization **does not** come at the cost of overwriting prior forgetting, specifically with the weights of the forget model being overwritten.

4.2 TOFU

In our second experiment, we evaluate our method on TOFU (Maini et al., 2024). For p and q, we use the LLaMA 3.2 1B and 3B *retain90* and *full* models (for retain and forget, respectively), and



(a) Verbatim memorization favors smaller p, q; Q&A favors larger p, q.

(b) A broad range of hyper-parameter settings balance over- and under-unlearning.

Figure 2: Analysis of Model Scaling and Over- or Under- Unlearning on MUSE

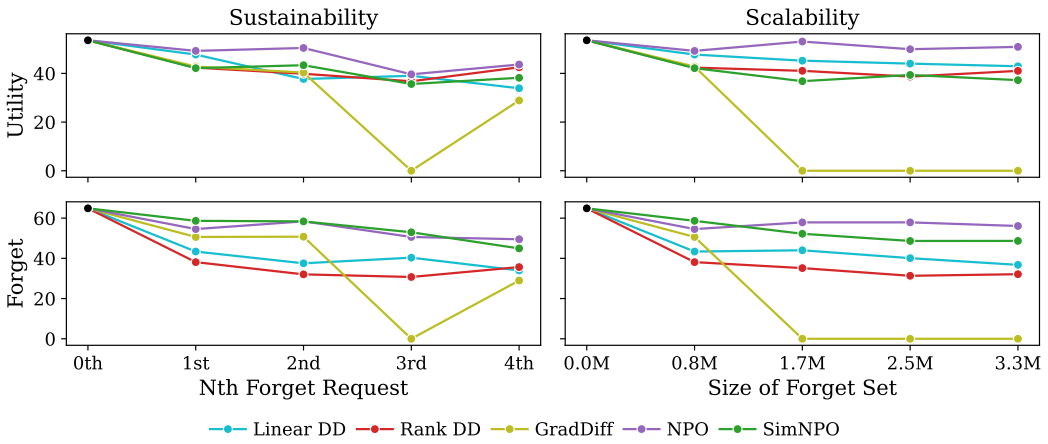


Figure 3: MUSE Scaling and Sustainability. The left column is sustainability - consecutive forget sets of the same size - and the right column is scaling, increasingly large forget sets. The top row is utility on the retain set, while the bottom row is the utility on the **original** forget set. We want the lines to stay flat after the first point on both rows, with the lines curving upwards on the bottom row being undesirable.

use the LLaMA 3.1 8B model as P. In Table 1, we find that with alphas slightly larger than 1, our method achieves an **almost perfect approximation of the retrain model**—including near perfect privacy scores. Performance gaps between the 1B and 3B model are much smaller compared to the 1.3B and 2.7B used in MUSE. Further details about the setup are provided in Appendix B.2

5 Beyond Text

One benefit of our method is its generality, i.e., it can be applied to any setting where samples are drawn from some distribution P and data exists to estimate p and q. In a final set of experiments we explore the extent to which our method is effective in domains beyond text by applying it to image generation.

We begin with the setup of Esser et al. (2021) and augment the sampling in latent space per equations 1 and 2. We estimate p and q using data from the train split of ImageNet which are associated with the dog synset. Specifically, we randomly assign half the descendants from the dog synset to the forget set F and the other half to the retain set R. We fine-tune the class-conditional ImageNet checkpoint from Esser et al. (2021) on F and R to estimate p and q, respectively. We then sample images from the model configured without any divergence decoding (Baseline) and with various

Table 1: TOFU Results

Method	Config	Agg. ↑	Mem. ↑	Priv. ↑	Utility ↑
Target	Full	0.02	0.01	0.38	1.00
Retrain	Retain90	0.78	0.53	1.00	1.03
Linear DD 1B	α=1.5	0.78	0.56	**0.95**	**1.00**
Linear DD 3B	α=1.2	0.79	0.56	**0.98**	**1.00**
Rank DD 1B	topk=20	**0.85**	**0.80**	0.81	0.95
Rank DD 3B	topk=20	**0.85**	**0.86**	0.77	0.93
DPO	lr=4e-6, epoch=2	0.32	0.21	0.39	0.46
GradAscent	lr=2e-6, epoch=3	0.63	0.51	0.61	0.86
GradDiff	lr=2e-6, epoch=3	0.63	0.52	0.62	0.85
NPO	lr=4e-6, epoch=2	0.67	0.57	0.68	0.81
RMU	lr=8e-7, epoch=4	0.67	0.60	0.74	0.68

Note: Agg. is the harmonic mean of Mem., Priv., and Utility. Each of these is itself the harmonic mean of several tests. The top two entries per column are boldfaced. See Appendix F of Dorna et al. (2025) for details on the construction of these metrics.

linear and rank-based setups. Following prior work (e.g., Heusel et al., 2017), we measure the quality of the generated images using the Fréchet Inception Distance (FID).

Ideally, samples from the model would no longer exhibit image semantics associated with the data in the forget set F, while retaining high perceptual quality relative to the retain set R. We assess performance by computing the FID between three pairs of data: (i) baseline images from the retain set and generated images using classes from the retain set (FID(B_R,G_R)), (ii) baseline images from the forget set and generated images using classes from the forget set (FID(B_F,G_F)), and (iii) baseline images from the retain set and generated images using classes from the forget set (FID(B_R,G_F)).

Efficacy in this setting preserves perceptual quality relative to the retain set, i.e., low FID(B_R,G_R) and low FID(B_R,G_F), while increasing the distance between the forget set and images generated based on those classes, i.e., high FID(B_F,G_F). In Table 2, we present FID statistics for a variety of decoding setups. For the linear setup, an $\alpha = 1$ seems to work well, e.g., a roughly 33% increase in FID(B_F,G_F) with only a 5% increase in FID(B_R,G_R) relative to the baseline. In contrast, the topk based methods appear to require much larger values of k to be effective.

Table 2: Quality of images generated using various divergence decoding setups.

Method	Config	FID(B_R,G_R) ↓	FID(B_F,G_F) ↑	FID(B_R,G_F) ↓
Baseline	—	18.2	18.0	30.1
Linear	$\alpha = 1$	19.2	24.1	27.3
Linear	$\alpha = 2$	20.5	28.7	26.8
Linear	$\alpha = 5$	22.8	31.6	25.8
Linear	$\alpha = 10$	22.6	31.4	25.3
Rank	topk=20	19.1	20.0	29.2
Rank	topk=100	20.1	22.1	28.7
Rank	topk=250	21.1	28.1	26.0

6 A TIME-UNLEARNING BENCHMARK

Existing unlearning methods—combined with improved data sanitation—are generally effective at eliminating the generation of toxic or copyrighted content. However, one of the most pressing motivations for unlearning arises in finance, where backtesting strategies on historical data is hindered by look-ahead bias, i.e., overly optimistic estimates of performance due to the model having been trained on data from the backtest period. Ideally, one would be able to leverage unlearning methods

to enable simulation of frontier models that had been trained to different knowledge cutoffs, thereby alleviating concerns about look-ahead bias.

To support the development of such methods, we propose a benchmark for measuring time-based unlearning by constructing a tractable proxy dataset which reflects contemporaneous events. Specifically, we download questions from Kalshi, a large, public prediction market. We retain all "one-off" or "two-off" questions (based on the series marker), which provide concise representations of events that were significant at the time or rumors that were trending at the time. In addition, we take the probability distribution **one hour** after the market has been created as our estimate of when the initial probabilities have converged and the market is liquid. Table 3 provides examples of questions, all of which were relevant at the time.

Table 3: Example prediction questions included in the benchmark. Correct answers appear in bold.

Question	Open Date	Close Date	Answer Choices
ChatGPT-5 revealed in 2024?	2023-12-13	2025-01-01	Revealed, **Not revealed**
Which team will draft Bronny James?	2024-05-03	2024-06-28	Undrafted, Portland Trail Blazers, ... **Los Angeles Lakers**
Will Bitcoin hit $100k again in 2024?	2024-12-11	2024-12-11	**Yes**, No

Importantly, these questions are inherently counterfactual in nature: they concern both events that **did** occur and those that **did not**. For evaluation, we pose each question as if it was the day market opened, and we score as if we spent $1 on contracts based on the recommendation of the model. Therefore, if the market is efficient, trading without any information edge (e.g., random guessing or always selecting the favorite) yields an expected average profit per trade of $0. Prior literature examining look-ahead bias in LLMs (e.g., Glasserman & Lin, 2023; Sarkar & Vafa, 2024) suggests estimates of model performance based on in-sample data will be overly optimistic. In Table 4, we find exactly this pattern, where apparent predictive ability in-sample vanishes out-of-sample.

Model	Pre-Cutoff	Post-Cutoff	Post-Release
GPT-4.1 mini	0.728 (n=222)	-0.006 (n=337)	**-0.144** (n=250)
GPT-4.1	**1.513** (n=222)	0.067 (n=337)	-0.064 (n=250)
o4-mini	**1.026** (n=222)	0.152 (n=337)	0.023 (n=250)
Claude 3.5 Haiku	**0.938** (n=248)	0.298 (n=7)	-0.027 (n=554)
Claude 3.7 Sonnet	**1.106** (n=256)	0.007 (n=248)	0.062 (n=305)
Claude 3.7 Sonnet - Extended Thinking	**1.194** (n=256)	0.038 (n=248)	-0.053 (n=305)
Llama 3.1 8B Instruct Turbo	0.626 (n=186)	0.010 (n=53)	**-0.163** (n=570)
Llama 3.1 70B Instruct Turbo	1.010 (n=186)	0.091 (n=53)	-0.136 (n=570)
Llama 3.1 405B Instruct Turbo	**1.300** (n=186)	0.047 (n=53)	**-0.254** (n=570)
DeepSeek V3	**1.147** (n=253)	**-0.201** (n=76)	-0.042 (n=480)
DeepSeek R1	**1.342** (n=253)	0.025 (n=189)	-0.002 (n=367)

Table 4: Model performance across different time periods. Average profit per trade from buying the model's chosen answer. Numbers significantly different from zero ($p < 0.05$) are bolded. Models exhibit significant look-ahead bias pre-cutoff and limited out-of-sample trading performance.

We consider the **diversity** of questions on our test a major strength. We don't intend to provide a forget corpus or for users to finetune directly on the questions. Instead, we intend for this to be used a general benchmark of any method that claims to reduce all look ahead bias.

7 LIMITATIONS

One limitation is the potential increase in **inference-time cost** due to running the small models in tandem with the large model. Let N denote the number of parameters in the large model and n the

number of parameters in each small model. Measured in FLOPs Kaplan et al. (2020), the inference cost scales from

$$2N \longrightarrow 2(N + 2n).$$

Additionally, let d_r and d_f be the sizes of the retain and forget datasets (in tokens), let e_N and e_n be the number of epochs the large and small models are trained for, respectively, and let I be the number of inference tokens. Hence, we want to know after how many inference tokens does it become more costly to use DD over another method, **assuming both work equally well**. Considering one of the simplest unlearning methods, Gradient Ascent (Jang et al., 2023) **without any kind of regularizer**, DD becomes more costly once:

$$\begin{aligned} 6ne_n(d_r + d_f) + 2(N + 2n)I &\geq 6Ne_N(d_f) + 2NI \\ I &\geq \frac{3Ne_N d_f}{2n} - \frac{3e_n(d_r + d_f)}{2} \end{aligned}$$

A second limitation is that DD does not erase internal representations; it only constrains outputs at decode time. This makes it unsuitable for preventing toxic or copyrighted generations in **open-weight settings**, since releasing the forget model's weights could reveal sensitive information. In addition, it would also be a privacy risk to allow users to access the topk logits or log probabilities from an API, since that could also reveal sensitive information.

A third limitation lies in the method's **sensitivity to instruction-tuning**. For instance, when unlearning financial knowledge, the model may generate stock recommendations in the format:

"**1. {firm name}:**"

If the smaller models anticipate a different structure (e.g., a ticker symbol or bullet marker after the '1.'), the divergence in logits at the critical step may be diluted or entirely noisy. Worse, if one small model aligns closely with the large model while the other does not, differences fail to cancel and can yield unstable or unintended outputs. Independent researchers adopting this method may therefore need to carefully re-tune instruction following behavior using publicly available datasets after modifying training mixtures, while in house researchers may not find this to be a problem.

8 Conclusion

In this work, we introduce an inference-time unlearning paradigm for selectively removing information from Large Language Models without costly retraining or fine-tuning of the base model. Our method, Divergence Decoding, leverages smaller, specialized models to guide text generation away from undesirable content at the point of inference. Our experiments demonstrate three key contributions. First, our approach is highly effective, significantly reducing the model's ability to recall both verbatim and semantic knowledge from a designated "forget set." Second, by confining training to small secondary models, our method offers a dramatically more efficient and scalable solution for machine unlearning, reducing computational overhead by orders of magnitude compared to existing techniques. Finally, because the weights of the large base model remain untouched, our method excels at utility preservation, maintaining performance on general knowledge benchmarks even as the number of unlearning requests grows. By providing a practical, low-cost, and effective solution to a critical challenge in AI safety and privacy, divergence decoding can potentially enable more responsible and adaptable deployment of large-scale language models.

Reproducibility Statement

We took care to modify OpenUnlearning as little as possible, and have details about our setups for MUSE and TOFU in Appendix B. All code to reproduce this work is available at the following anonymous repository link: https://anonymous.4open.science/r/inference-time-unlearning-iclr2026/

We will release fine-tuned models and data on Hugging Face after the review period.

Ethics Statement

In general, we intend unlearning to support beneficial use cases - for debiasing models, preventing toxic and copyrighted content generation, and legitimate research in domains such as finance. However, we acknowledge the approach could be misused to induce undesirable or harmful biases.

References

Josh Achiam, Steven Adler, and the OpenAI GPT-4 Team. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Karuna Bhaila, Minh-Hao Van, and Xintao Wu. Soft prompting for unlearning in large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4046–4056, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.204. URL https://aclanthology.org/2025.naacl-long.204/.

Louis Bourtoule, Armin W. Thomas, Fabian Pedregosa, Blake Sorensen, Robert West, Manuel Gomez Rodriguez, and Nicolas Papernot. Machine unlearning. *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 568–584, 2021. doi: 10.1109/SP40001.2021.00045.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. Large language models in machine translation. In Jason Eisner (ed.), *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 858–867, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/D07-1090/.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Martin Juan José Bucher and Marco Martini. Fine-tuned 'small' llms (still) significantly outperform zero-shot generative ai models in text classification, 2024. URL https://arxiv.org/abs/2406.08660.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Xiaodong Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *USENIX Security Symposium*, 2021. URL https://api.semanticscholar.org/CorpusID:229156229.

DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024. URL https://arxiv.org/abs/2405.04434.

DeepSeek-AI. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramon Huerta, and Ivan Vulić. Undial: Self-distillation with adjusted logits for robust unlearning in large language models, 2024. URL https://arxiv.org/abs/2402.10052.

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. URL https://arxiv.org/abs/2506.12618.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms, 2023. URL https://arxiv.org/abs/2310.02238.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, June 2021.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082/.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Neurips Safe Generative AI Workshop 2024*, 2024. URL https://openreview.net/forum?id=pVACX02m0p.

Chongyang Gao, Lixu Wang, Kaize Ding, Chenkai Weng, Xiao Wang, and Qi Zhu. On large language model continual unlearning, 2025. URL https://arxiv.org/abs/2407.10223.

Paul Glasserman and Caden Lin. Assessing look-ahead bias in stock return predictions generated by gpt sentiment analysis, 2023. URL https://arxiv.org/abs/2309.17322.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, and the rest of the Llama 3 team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. URL https://arxiv.org/abs/2306.11644.

J.M. Hammersley and D.C. Handscomb. *Monte Carlo Methods*. Methuen's monographs on applied probability and statistics. Chapman and Hall, 1965. ISBN 9789400958203. URL https://books.google.com/books?id=P3FqAAAAMAAJ.

Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A. Lemley, and Percy Liang. Foundation models and fair use. *Journal of Machine Learning Research*, 24(400):1–79, 2023. URL http://jmlr.org/papers/v24/23-0569.html.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Geoffrey E. Hinton. Products of experts. 1999. URL https://api.semanticscholar.org/CorpusID:15059668.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama guard: Llm-based input-output safeguard for human-ai conversations, 2023. URL https://arxiv.org/abs/2312.06674.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models, 2023. URL https://openreview.net/forum?id=zAxuIJLb38.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2023. URL https://arxiv.org/abs/2211.17192.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Zhenqi Zhao, Ariel Herbert-Voss, Cort B. Breuer, Samuel Marks, Oam Patel, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, Kallol Krishna Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, Ponnurangam Kumaraguru, Uday Tupakula, Vijay Varadharajan, Ruoyu Wang, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024. URL https://arxiv.org/abs/2403.03218.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*, 2024.

Andrei Muresanu, Anvith Thudi, Michael R. Zhang, and Nicolas Papernot. Unlearnable algorithms for in-context learning, 2024. URL https://arxiv.org/abs/2402.00751.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Trans. Intell. Syst. Technol.*, July 2025. ISSN 2157-6904. doi: 10.1145/3749987. URL https://doi.org/10.1145/3749987.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 40034–40050. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/pawelczyk24a.html.

Branislav Pecher, Ivan Srba, and Maria Bielikova. Comparing specialised small and general large language models on text classification: 100 labelled samples to achieve break-even performance, 2025. URL https://arxiv.org/abs/2402.12819.

Suproteem Sarkar and Keyon Vafa. Lookahead bias in pretrained language models. Available at SSRN: https://ssrn.com/abstract=4754678 or http://dx.doi.org/10.2139/ssrn.4754678, June 2024.

Mrinank Sharma, Meg Tong, Jesse Mu, Jerry Wei, Jorrit Kruthoff, Scott Goodfriend, Euan Ong, Alwin Peng, Raj Agarwal, Cem Anil, Amanda Askell, Nathan Bailey, Joe Benton, Emma Bluemke, Samuel R. Bowman, Eric Christiansen, Hoagy Cunningham, Andy Dau, Anjali Gopal, Rob Gilson, Logan Graham, Logan Howard, Nimit Kalra, Taesung Lee, Kevin Lin, Peter Lofgren, Francesco Mosconi, Clare O'Hara, Catherine Olsson, Linda Petrini, Samir Rajani, Nikhil Saxena, Alex Silverstein, Tanya Singh, Theodore Sumers, Leonard Tang, Kevin K. Troy, Constantin Weisser, Ruiqi Zhong, Giulio Zhou, Jan Leike, Jared Kaplan, and Ethan Perez. Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming, 2025. URL https://arxiv.org/abs/2501.18837.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models, 2023.

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. 2024. URL https://arxiv.org/abs/2407.06460.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TArmA033BU.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed.*, 2017.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. Ethical and social risks of harm from language models, 2021. URL https://arxiv.org/abs/2112.04359.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*, 2023.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning, 2024. URL https://arxiv.org/abs/2404.05868.

A Theoretical Results

A.1 Connection to Product of Experts

Hinton (1999) introduced the Product of Experts (PoE) framework whereby n probability models are multiplicatively combined into a single model. Let the i-th expert be denoted by $f_i(x|\theta_i)$, then a PoE model R comprised of n experts is given by,

$$R(x|\theta_1, ..., \theta_n) = \frac{1}{Z} \prod_{i=1}^{n} f_i(x|\theta_i), \tag{6}$$

where Z is a normalization constant. To highlight the connection between divergence decoding and PoE, recall Eq. 1:

$$\hat{l}_Q(x_{<t}) = l_P(x_{<t}) + \alpha \cdot [l_q(x_{<t}) - l_p(x_{<t})].$$

In Eq. 1, a given model M has logits which are equal to the log-probabilities up to an additive constant which depends on the token sequence prefix $x_{<t}$ but not the token x_t, i.e.,

$$l_M(x_{<t}) = \log M(x_t|x_{<t}) + C_M(x_{<t}). \tag{7}$$

Substituting Eq. 7 into Eq. 1 for each model, gathering the constants, and performing some algebra reveals the link to PoE:

$$\begin{aligned}
\log \widehat{Q}(x_t|x_{<t}) &= \log P(x_t|x_{<t}) + \alpha \cdot [\log q(x_t|x_{<t}) - \log p(x_t|x_{<t})] + C \\
\widehat{Q}(x_t|x_{<t}) &\propto \exp\big(\log P(x_t|x_{<t}) + \alpha \cdot [\log q(x_t|x_{<t}) - \log p(x_t|x_{<t})]\big) \\
&\propto P(x_t|x_{<t}) \cdot q(x_t|x_{<t})^{\alpha} \cdot p(x_t|x_{<t})]^{-\alpha} \\
&\propto P(x_t|x_{<t}) \cdot \left[\frac{q(x_t|x_{<t})}{p(x_t|x_{<t})}\right]^{\alpha}.
\end{aligned}$$

B Experiments

B.1 Detailed muse setup and Analysis

We finetune the LlaMA models using the **AdamW Torch optimizer** and a **cosine scheduler** for **10** epochs. We set the learning rate such that the loss approximately halves over the course of training.

We sweep $\alpha \in \{0.5, 0.6, \ldots, 1.5\}$ and top-$k \in \{1, 5, 20, 50, 100, 200, 500, 1000\}$ for the LLaMA models, and at $\alpha \in \{5, 10, \ldots 30\}$ and top-$k \in \{1, 2, 3, 5, 10\}$ for the trigram models. We choose the most optimal point as the point closest in euclidean distance to Retrain. We find that in general, rank DD outperforms on verbatim memorization while linear DD outperforms on Q&A knowledge.

Table 5: Configuration MUSE

Model	Initial LR	Best Verbatim	Best Q&A
Stupid Backoff Trigram		TopK=1	Alpha=10
princeton-nlp/Sheared-LLaMA-1.3B	5e-5	TopK=100	Alpha=0.8
princeton-nlp/Sheared-LLaMA-2.7B	4e-5	TopK=200	Alpha=1.0

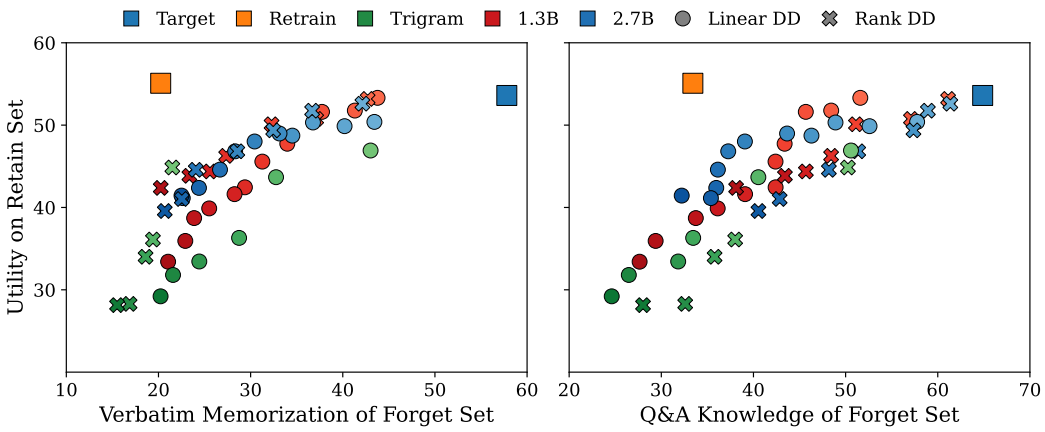


Figure 4: All hyper-parameter and model size configurations

For the other methods, we use the default settings provided by OpenUnlearning

Table 6: MUSE Configurations

Method	Epochs	Method-Specific Hyperparameters
GradDiff	1*	$\alpha = 1.0,\ \gamma = 1.0$
NPO	10	$\beta = 0.1,\ \alpha = 1.0,\ \gamma = 1.0$
SimNPO	10	$\delta = 0,\ \beta = 4.5,\ \alpha = 1.0,\ \gamma = 0.125$

Default hyperparameters: batch size = 32, learning rate = 1×10^{-5}, warmup epochs = 1, weight decay = 0.01, retain loss = NLL. * For GradDiff, the 1 epoch setting is the only deviation from the defaults.

B.2 Detailed TOFU Setup

For the other methods, we grid search learning rates $\{5 \times 10^{-7}, 8 \times 10^{-7}, 1 \times 10^{-6}, 2 \times 10^{-6}, 3 \times 10^{-6}, 4 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}\}$ and epochs from 1 to 10. Below, we summarize the default hyperparameters provided by OpenUnlearning.

Table 7: TOFU Default Configurations (defaults apply unless noted)

Method	Method-Specific Hyperparameters
DPO	$\beta = 0.1,\ \alpha = 1.0,\ \gamma = 1.0,$ retain loss = NLL
GradAscent	N/A
GradDiff	$\alpha = 1.0,\ \gamma = 1.0,$ retain loss = NLL
NPO	$\beta = 0.1,\ \alpha = 1.0,\ \gamma = 1.0,$ retain loss = NLL
RMU	$\alpha = 1.0,\ \gamma = 1.0,$ steering coef = 2, retain loss = Embed Diff

Default (shared) hyperparameters: batch size = 32, warmup epochs = 1, weight decay = 0.01.