UCD: Unlearning in LLMs via Contrastive Decoding

Vinith Suriyakumar MIT
Ayush Sekhari* Boston University
Ashia Wilson* MIT

Abstract

Machine unlearning aims to remove specific information, e.g. sensitive or undesirable content, from large language models (LLMs) while preserving overall performance. We propose an inference-time unlearning algorithm that uses contrastive decoding, leveraging two auxiliary smaller models, one trained without the forget set and one trained with it, to guide the outputs of the original model using their difference during inference. Our strategy substantially improves the tradeoff between unlearning effectiveness and model utility. We evaluate our approach on two unlearning benchmarks, TOFU and MUSE. Results show notable gains in both forget quality and retained performance in comparison to prior approaches, suggesting that incorporating contrastive decoding can offer an efficient, practical avenue for unlearning concepts in large-scale models.

1 Introduction

Large Language Models (LLMs) achieve impressive general capabilities thanks to massive training datasets and compute. However, these capabilities raise significant safety and security concerns, including copyright violations (Karamolegkou et al., 2023), harmful content generation, and retention of dangerous knowledge (e.g., bioweapon instructions) (Shevlane et al., 2023). Retraining models to address these issues by excluding problematic data is impractical at scale. This has led to growing interest in efficient methods for *machine unlearning*, which aim to remove specific information from trained models without retraining.

The field of machine unlearning began with a focus on removing the influence of specific training data points from trained machine learning models (Cao & Yang, 2015; Bourtoule et al., 2021; Neel et al., 2021; Sekhari et al., 2021; Ghazi et al., 2023; Suriyakumar & Wilson, 2022). This initial motivation arose primarily from compliance with emerging privacy regulations, such as the EU's General Data Protection Regulation (GDPR) (Voigt & Von dem Bussche, 2017) and the California Consumer Privacy Act (CCPA) (CCP, 2018), both of which enforce the *Right to be Forgotten*. More recently, researchers concerned with AI safety have broadened the scope of machine unlearning to also include removing unwanted or harmful knowledge from large language models (Li et al., 2024; Barez et al., 2025; Zhang et al., 2023b).

*Equal Advisory Contribution.

So far, two broad classes of unlearning algorithms have been proposed for LLM unlearning: *finetuning-based* approaches and *representation-engineering* approaches. Finetuning algorithms define an objective to represent "unlearning" and optimize it using samples of data to be forgotten (i.e., the *forget set*). The canonical example is gradient ascent, which maximizes empirical loss on the forget set. Extensions of this approach incorporate additional loss terms to maintain model utility (Jang et al., 2022; Yao et al., 2023a; Chen & Yang, 2023; Schwarzschild et al., 2024) or modify alignment procedures, such as direct preference optimization (Rafailov et al., 2024). Representation-engineering methods propose objectives to modify internal representations of the model with respect to the forget set, typically by projecting them onto random or orthogonal subspaces (Li et al., 2024). A shortcoming of both of these classes of methods is that they are expensive to run and suffer from poor forget-utility tradeoffs (Shi et al., 2024).

Motivated by recent advances in inference-time methods that improve reasoning and alignment without extensive retraining, we propose **Unlearning via Contrastive Decoding (UCD)**, a novel inference-time unlearning algorithm inspired by contrastive decoding principles (Li et al., 2023). UCD leverages two small auxiliary models, one trained exclusively on the forget set and another trained on the retain set, allowing it to effectively remove undesirable knowledge at inference (Figure 1). This approach significantly improves the forget-utility tradeoff and sets new state-of-the-art benchmarks on established unlearning datasets (TOFU and MUSE News). Additionally, due to its computational efficiency, UCD enables practical unlearning even on extremely large models such as Llama2-70B, a task previously infeasible with existing approaches. **Our main contributions are:**

- **Contrastive Decoding-Based Unlearning:** We introduce UCD, an efficient inference-time unlearning algorithm utilizing two auxiliary models trained separately on forget and retain data. Whenever it is possible to obtain a clean model trained solely on the forget set, or when the data is sufficiently separable to allow targeted fine-tuning, UCD can be easily applied.
- **Superior Forget-Utility Tradeoff:** UCD significantly outperforms existing methods on standard machine unlearning benchmarks (TOFU, MUSE), achieving forget performance indistinguishable from retraining and improved utility due to contrastive decoding's enhanced text quality.
- **Scalability to Significantly Larger Models:** Unlike existing weight-modifying unlearning methods constrained by computational costs, UCD demonstrates practical inference-time unlearning on significantly larger models, including Llama2-13B and Llama2-70B, only requiring 2 L40s for unlearning on Llama2-13B and 4 NVIDIA H200s for unlearning on Llama2-70B. Whereas all pre-existing baselines required at least 2 A100s for unlearning on Llama2-13B and are infeasible on Llama2-70B on 8 H200s.

2 Background and Related Work

Machine unlearning. Our work builds on a growing body of research on machine unlearning (Bourtoule et al., 2021; Nguyen et al., 2022; Cao & Yang, 2015; Gupta et al., 2021;

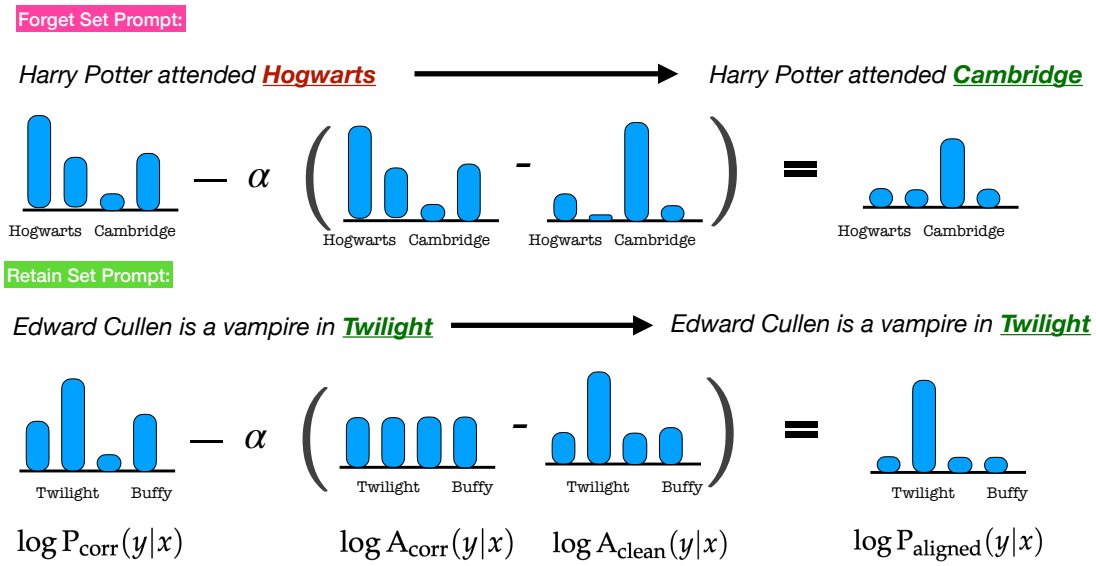

Figure 1: Illustration of contrastive decoding at inference time in UCD. In the top row, we prompt the model with a sentence from our forget set corpus. The first distribution represents the original model we would like to unlearn from. The difference between our two auxiliary models guides the distribution to suppress the information related to Harry Potter. Meanwhile, in the second row, on a prompt we would like to retain the difference remains small leaving the output unaffected.

Suriyakumar & Wilson, 2022; Sekhari et al., 2021; Ghazi et al., 2023; Kurmanji et al., 2023; Lev & Wilson, 2024; Łucki et al., 2024), which aims to develop methods that efficiently modify trained machine learning models to forget specific portions of their training data. In the case of classical discriminative models, the motivation often stems from privacy concerns, particularly the need to protect individuals whose data may have been used during training. A major driver behind this line of research was the introduction of Article 17 of the European Union's General Data Protection Regulation (GDPR), which codifies an individual's "right to be forgotten"(European Union, 2016). Various other legislations have followed including California Consumer Privacy Act (CCPA), Canada's proposed Consumer Privacy Protection Act (CPPA)), and more recently in Australia (Karp, 2023). More recently, the scope of machine unlearning has expanded to include modern generative AI models, which pose additional challenges such as the potential reproduction of copyrighted material, generation of harmful or explicit content, and leakage of sensitive training data (Zhang et al., 2023a; Carlini et al., 2021).

Unlearning and alignment in LLMs. Machine unlearning for Large Language Models (LLMs) has emerged as a rapidly growing area of research (Liu et al., 2024a; Jang et al., 2022; Kumar et al., 2022; Zhang et al., 2023b; Pawelczyk et al., 2023; Yao et al., 2023a; Zhang et al., 2024; Wang et al., 2024; Jia et al., 2024; Lu et al., 2022; Liu et al., 2024b; Ishibashi & Shimodaira, 2023; Thaker et al., 2024; Kadhe et al., 2024; Fan et al.). Given the inherent

difficulty of exact unlearning, most existing approaches rely on approximate methods such as fine-tuning and representation engineering (Yao et al., 2023a; Eldan & Russinovich, 2023; Jia et al., 2024; Zhang et al., 2024; Li et al., 2024; Ilharco et al., 2022; Liu et al., 2022) or prompt-based and in-context learning techniques (Thaker et al., 2024; Pawelczyk et al., 2023; Liu et al., 2024a). Numerous benchmarks and evaluations have been developed to measure the effectiveness of these heuristical unlearning algorithms (Maini et al., 2024; Shi et al., 2024; Li et al., 2024). We also highlight that test-time methods have started to gain prominence in LLM alignment, specifically using token-level rewards to guide generations (Xu et al., 2024). We view our work as a similar family of methods where UCD provides a new reward designed for unlearning and representing the next token distribution if the model was trained without the forget set.

We defer a detailed description of the baselines that we compare to in our experiments, as well as additional related work on model-editing for unlearning, to Appendix A.

3 UCD: Unlearning via Contrastive Decoding

This work focuses on the problem of *machine unlearning* for large language models (LLMs). Given an initial model $\mathrm{P}_{\mathrm{corr}}(y|x)$, referred to as the *corrupted* or *reference model*, that has been trained on a dataset $\mathcal{D} = (x_i, y_i)_{i=1}^n$, the central objective of machine unlearning is to effectively erase all information related to a designated subset of the dataset $\boldsymbol{\mathcal{D}}_{\mathbf{forget}} \subseteq \mathcal{D}$, termed the *forget set*, while preserving the model's performance on the remaining subset $\boldsymbol{\mathcal{D}}_{\mathbf{retain}}$, known as the *retain set*.

We approach the unlearning problem by leveraging auxiliary models to adjust the sampling distribution of the reference model. In particular, suppose there exists some public dataset $\mathcal{D}_{\texttt{pretrain}}$ that does not contain $\boldsymbol{\mathcal{D}}_{\mathbf{forget}}$, and a clean base model A trained on $\mathcal{D}_{\texttt{pretrain}}$. Using this base model, we will first train two auxiliary models $\mathrm{A}_{\mathrm{corr}}$ and $\mathrm{A}_{\mathrm{clean}}$ by separately fine-tuning A on the forget set $\boldsymbol{\mathcal{D}}_{\mathbf{forget}}$ and the retain set $\boldsymbol{\mathcal{D}}_{\mathbf{retain}}$, respectively. Without loss of generality, we assume the base model A is significantly smaller than the reference model P, making the fine-tuning process to obtain $\mathrm{A}_{\mathrm{corr}}$ and $\mathrm{A}_{\mathrm{clean}}$ substantially less resource-intensive compared to directly fine-tuning or retraining P. For example, A could be a Llama2-7B model, while P could be a much larger Llama2-70B model, thus significantly reducing computational requirements.

Unlearning via Contrastive Decoding (UCD). We utilize the contrastive decoding approach of Li et al. (2023) to define the logits for the returned model $\mathcal{L}_{\texttt{aligned}}$, i.e. we set

$$\log \mathrm{P}_{\mathrm{aligned}}(y|x) \leftarrow \log \mathrm{P}_{\mathrm{corr}}(y|x) - \alpha \cdot (\log \mathrm{A}_{\mathrm{corr}}(y|x) - \log \mathrm{A}_{\mathrm{clean}}(y|x)) \tag{1}$$

where $\alpha > 0$ denotes a hyper-parameter (set to 0.1 in Li et al. (2023)). Correspondingly, once we have the logits corresponding to $\mathrm{P}_{\mathrm{aligned}}$, we can generate next token using either:

- **Greedy Decoding (e.g. max-sampling)**: Given an input sequence x, select the next token y by choosing the one with the highest predicted probability according to model $\mathrm{P}_{\mathrm{aligned}}$: $y = \arg\max_{y'} \log \mathrm{P}_{\mathrm{aligned}}(y'|x)$.
- **Stochastic Decoding (e.g. nucleus sampling)**: Given an input sequence x, randomly select the next token y based on normalized distribution given by the subset of tokens

from $\mathrm{P}_{\text{aligned}}$ whose cumulative probability exceeds some threshold p, where p controls the amount of randomness.

Our unlearning update (1) modifies the logits of the reference model P_{corr} using the difference between the logits of the auxiliary models A_{corr} and $\mathrm{A}_{\text{clean}}$.[1] Recall that A_{corr} is fine-tuned on D_{forget}, while $\mathrm{A}_{\text{clean}}$ is fine-tuned on D_{retain}. The contrastive signal, defined as $\Delta_A(y \mid x) := \log \mathrm{A}_{\text{corr}}(y \mid x) - \log \mathrm{A}_{\text{clean}}(y \mid x)$, captures how much more strongly the forget-tuned model A_{corr} prefers next-token y for a given prompt x when compared to the retain-tuned model $\mathrm{A}_{\text{clean}}$.

This contrastive signal forms the basis of our approach: we can unlearn by simply adjusting the logits of the reference model using the difference in token preferences between auxiliary models trained with and without the forget set. For illustration, if we prompt the model with a query about a data sample that should be erased (i.e. $(x, y) \in D_{\text{forget}}$), both $\Delta_A(y \mid x)$ and $\log \mathrm{P}_{\text{corr}}(y \mid x)$ are likely to be high. Thus, the update in (1) reduces the logit for y, thereby lowering its probability in the generative process and suppressing this information. More generally, when $\Delta_A(y \mid x)$ is large and positive, i.e., A_{corr} favors y significantly more than $\mathrm{A}_{\text{clean}}$, the update decreases $\log \mathrm{P}_{\text{corr}}(y \mid x)$ and thereby reduces the likelihood of generating y. Conversely, when $\Delta_A(y \mid x)$ is large and negative, indicating that $\mathrm{A}_{\text{clean}}$ prefers y more than A_{corr}, the update increases $\log \mathrm{P}_{\text{corr}}(y \mid x)$ and thereby increases the probability of y.

Unlearning via Contrastive Suppression (UCS). While UCD can both increase or decrease the probability of outputting various tokens in P_{corr}, depending on the sign of the contrastive signal $\Delta(y \mid x)$, in various cases, we may want to be more conservative and only make a relative decrease in logits (instead of both increasing and decreasing them using the auxiliary models). Towards that end, we also propose an update step that clips off the impact of contrastive decoding when the contrastive single is negative:

$$\log \mathrm{P}_{\text{aligned}}(y \mid x) \leftarrow \log \mathrm{P}_{\text{corr}}(y|x) - \max\{\log \mathrm{A}_{\text{corr}}(y|x) - \log \mathrm{A}_{\text{clean}}(y|x), 0\}$$

where $\alpha > 0$ is a hyperparameter. Again, after computing the new logits, we can sample using a greedy or stochastic decoding approach.

3.1 Why is our Contrastive Decoding Approach Effective for Unlearning?

We offer an initial intuition for the potential effectiveness of UCD. Although we do not present this as a comprehensive explanation of the observed behavior, we believe it sheds light on some of the underlying dynamics at play. Throughout this section, let $\mathrm{P}_{\text{clean}}$ denote the model we would have obtained (corresponding to the corrupted model P_{corr}) if we had trained the given reference model without the forget set D_{forget}.

First, as a sanity check, observe that if the auxiliary models are chosen to be the same size as the underlying models, that is, $\mathrm{A}_{\text{clean}} = \mathrm{P}_{\text{clean}}$ and $\mathrm{A}_{\text{corr}} = \mathrm{P}_{\text{corr}}$, and we set $\alpha = 1$,

[1] While (1) represents the output of contrastive decoding by $\mathrm{P}_{\text{aligned}}$, we emphasize that no new model is computed; instead, only the logits—used to define the next-token distribution—are modified.

then:

$$\log \mathrm{P}_{\mathrm{corr}}(y \mid x) - \alpha \cdot (\log \mathrm{A}_{\mathrm{corr}}(y \mid x) - \log \mathrm{A}_{\mathrm{clean}}(y \mid x)) = \log \mathrm{P}_{\mathrm{clean}}(y \mid x). \tag{2}$$

In this special case, the contrastive differencing update in (1) exactly recovers the next token distribution corresponding to the model $\mathrm{P}_{\mathrm{clean}}$ that is retrained-from-scratch on the retain set. This illustrates, in idealized conditions, how our approach enables unlearning.

We now relax this strong equivalence assumption to examine more practical settings where the auxiliary models differ in scale or capacity from the underlying models.

Proposition 1. *Suppose that for any input prompt x, the auxiliary models $\mathrm{A}_{\mathrm{corr}}$ and $\mathrm{A}_{\mathrm{clean}}$ satisfy the relation:*

$$\log \mathrm{A}_{\mathrm{corr}}(y \mid x) - \log \mathrm{A}_{\mathrm{clean}}(y \mid x) \propto \log \mathrm{P}_{\mathrm{corr}}(y \mid x) - \log \mathrm{P}_{\mathrm{clean}}(y \mid x), \tag{3}$$

for any token $y \in \mathcal{Y}$, where $\mathrm{P}_{\mathrm{corr}}$ denotes the initial corrupted model, and $\mathrm{P}_{\mathrm{clean}}$ denotes the clean model (obtained by retraining-from-scratch without the retain set). Then, there exists a choice of α that is independent of y such that the contrastive decoding procedure in (1) ensures that $\mathrm{P}_{\mathrm{aligned}} \equiv \mathrm{P}_{\mathrm{clean}}$.

The assumption in (3) formalizes the intuition that small auxiliary models can generalize the token-level preference trends observed in large models, even if the magnitude of those preferences is not preserved. Specifically, (3) suggests that if there exist tokens for which the logit difference $\log \mathrm{P}_{\mathrm{corr}}(y \mid x) - \log \mathrm{P}_{\mathrm{clean}}(y \mid x)$ is large, indicating that the corrupted model strongly prefers token y compared to the clean model (and hence y should be suppressed), then a similar trend should be observable in the auxiliary models $\mathrm{A}_{\mathrm{corr}}$ and $\mathrm{A}_{\mathrm{clean}}$.

The proof is straightforward. Suppose the constant of proportionality in (3) is m, i.e.,

$$\log \mathrm{A}_{\mathrm{corr}}(y \mid x) - \log \mathrm{A}_{\mathrm{clean}}(y \mid x) = m \left(\log \mathrm{P}_{\mathrm{corr}}(y \mid x) - \log \mathrm{P}_{\mathrm{clean}}(y \mid x)\right). \tag{4}$$

Then, setting $\alpha = 1/m$ in the UCD update (1) ensures that $\mathrm{P}_{\mathrm{aligned}} = \mathrm{P}_{\mathrm{clean}}$, thereby recovering the target unlearning model exactly. While the strict proportionality in (3) may be too strong to hold exactly in practice, the UCD update remains effective when this relationship holds approximately. Specifically, suppose there exist constants $c_1, c_2 > 0$ such that for any token y with $\log \mathrm{P}_{\mathrm{corr}}(y \mid x) - \log \mathrm{P}_{\mathrm{clean}}(y \mid x) \geq 0$, we have:

$$c_1 \leq \frac{\log \mathrm{A}_{\mathrm{corr}}(y \mid x) - \log \mathrm{A}_{\mathrm{clean}}(y \mid x)}{\log \mathrm{P}_{\mathrm{corr}}(y \mid x) - \log \mathrm{P}_{\mathrm{clean}}(y \mid x)} \leq c_2. \tag{5}$$

In this case, choosing $\alpha \in [1/c_2, 1/c_1]$ approximately aligns $\mathrm{P}_{\mathrm{aligned}}$ with $\mathrm{P}_{\mathrm{clean}}$, suppressing undesirable completions associated with the forget set while boosting completions consistent with the retain set.

4 Experimental Setup

We evaluate UCD on three different tasks from two different unlearning benchmarks: Task of Fictitious Unlearning (TOFU) (Maini et al., 2024) and Machine Unlearning Six Ways

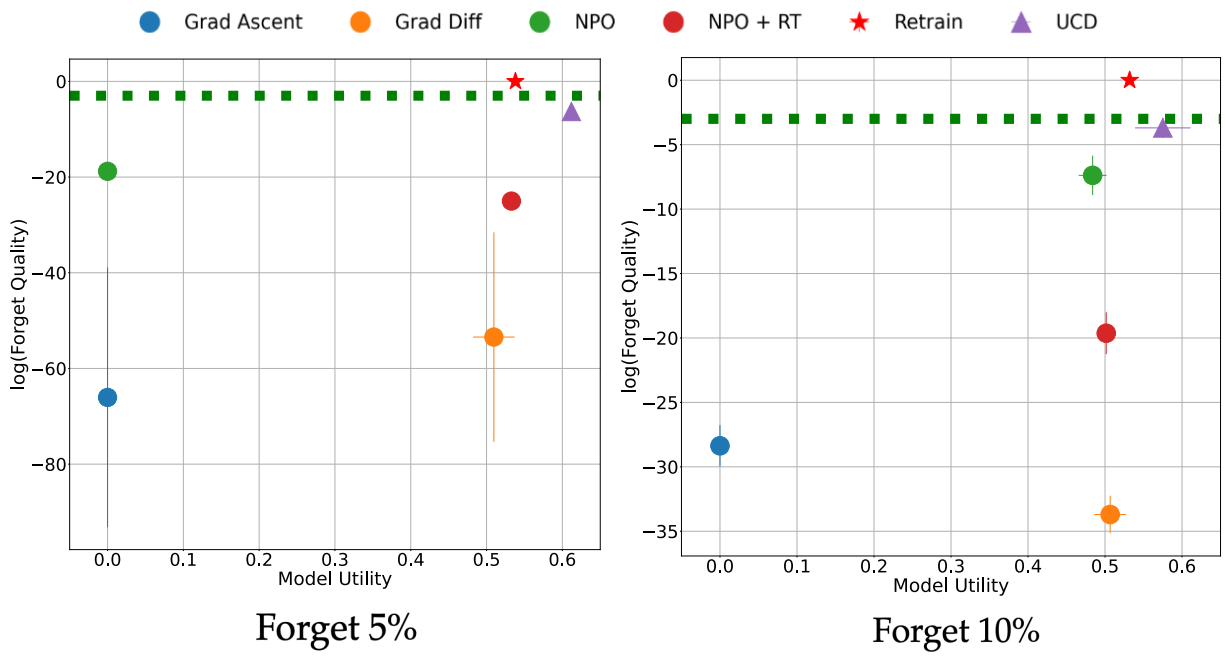

Figure 2: Forget quality versus model utility averaged over three random seeds for TOFU 5% (left) and TOFU 10% (right) on Llama2-13B. The dotted green line represents the forget quality (log(0.05)) algorithms must be greater than or equal to, to be considered indistinguishable from the retrained baseline. UCD (using Llama2-7B auxiliary models) achieves the best forget quality-model utility tradeoff.

Evaluation (MUSE) (Shi et al., 2024). All of our evaluations are on Llama2-13B as P and Llama2-7B as our auxiliary models, A. This is the first time, to our knowledge, that existing unlearning baselines have been studied on larger models than Llama2-7B. Below we describe the specific tasks from each benchmark and the metrics used to evaluate the unlearning methods.

TOFU. TOFU contains 200 GPT-4 generated author profiles, with 20 question-answer pairs for each author. The generated profiles were not contained in the pretraining data, resulting in a suitable setup for studying unlearning. We pick two tasks from TOFU: Forget 5% and Forget 10%, which represent forgetting 5% and 10% of the data, respectively. We evaluate the unlearning algorithms on these tasks according to four different sets of QA pairs: forget set, retain set, real world authors, and real world facts. More details about these sets can be found in the original TOFU paper (Maini et al., 2024). We focus on measuring log(*Forget Quality*) and *Model Utility* as described in Maini et al. (2024). Forget quality measures how indistinguishable the unlearned model is from the gold-standard retrained model. Indistinguishability is formalized as the p-value of a Kolmogorov-Smirnov test being above 0.05. Model utility measures the performance of the model on the retain set, real world authors, and real world facts sets. We report the additional metrics from TOFU of: ROUGE-L recall, probability, and truth ratio in the appendix.

Algorithm	VerbMem on $\mathcal{D}_{\text{forget}} \downarrow$	PrivLeak	KnowMem on $\mathcal{D}_{\text{forget}} \downarrow$	KnowMem on $\mathcal{D}_{\text{retain}} \uparrow$
Retrain	**20.99 ± 0.42**	**0.00 ± 0.00**	**38.08 ± 2.13**	**46.15 ± 1.49**
UCD	**20.5 ± 0.56**	**9.55 ± 6.65**	**36.38 ± 0.9**	**43.87 ± 1.37**
Grad Ascent	0.0 ± 0.0	58.97 ± 8.25	0.0 ± 0.0	0.0 ± 0.0
Grad Diff	0.0 ± 0.0	-23.41 ± 3.07	0.0 ± 0.0	0.0 ± 0.0
NPO + RT	1.02 ± 0.83	64.58 ± 3.22	28.78 ± 2.85	34.27 ± 2.16

Table 1: Forget quality (first three columns) versus model utility (last column) for MUSE News. UCD achieves the best forget quality-model utility tradeoff, almost approaching the retrained model.

MUSE. MUSE represents two different corpuses of text: news articles and books. The News task contains BBC articles after 2023, and the Books task contains all of the Harry Potter books. We focus on the News task in this work because we were unable to obtain a "clean" model for the Books task. Since we know the cutoff data for the Llama2 models this makes it easy to have clean models for the News task. Meanwhile, obtaining a clean model for Llama2 for the Books task would require pretraining a model from scratch. Similar to TOFU, we evaluate both the forget quality and model utility. For forget quality, we measure the verbatim memorization of the forget set (`VerbMem` on $\boldsymbol{\mathcal{D}_{\text{forget}}}$), the ability to infer membership in the training data (`PrivLeak` (Shi et al., 2023)), and knowledge retention via QA on the forget set (`KnowMem` on $\boldsymbol{\mathcal{D}_{\text{forget}}}$). Model utility is measured by knowledge retention via QA on the retain set (`VerbMem` on $\boldsymbol{\mathcal{D}_{\text{retain}}}$).

Training and Unlearning. For all three tasks, we compare our method against the following baselines: gradient ascent (Maini et al., 2024), gradient difference (Liu et al., 2022), negative preference optimization (NPO) (Zhang et al., 2024), and NPO with a retain loss (NPO + RT). All of these baselines are described in Appendix A and were run following the open-source implementations from both the TOFU and MUSE benchmarks. We pick this subset of methods out of the ones discussed based on their performance in prior works on the chosen tasks. We average all of the results for the baselines and our method over three random seeds. We use a range of compute depending on the algorithm. Specifically, going from two NVIDIA L40s with 48GB of VRAM to 8 NVIDIA H200s with 141GB of VRAM depending on out of memory errors encountered when running on smaller amounts of compute. We elaborate more on this need to use a range of compute and how UCD is much more efficient compute wise.

5 Results

5.1 UCD Improves Forgetting-Utility Tradeoff

UCD significantly outperforms the baselines across all three tasks. We report the best performing UCD model based on tuning of α over $\{0.01, 0.1, 0.5, 1.0\}$. As shown in Figure 2, for TOFU 5%, UCD achieves indistinguishability from the retrained model and also improves the model utility. We believe that the improvement in utility compared to the retrained

model can be attributed to the contrastive decoding approach. Numerous prior works show contrastive decoding improves text quality and diversity (Li et al., 2023; O'Brien & Lewis, 2023). We provide an example of how UCD successfully recovers the retrained model compared to all other baselines in Appendix B. For MUSE, UCD is the closest model to replicating the retrained model (Table 1). UCD overcomes issues of over unlearning / under unlearning (measured by the `PrivLeak` metric) and poor model utility discussed in the original paper.

5.2 Bootsrapping from Existing Unlearning Algorithms Improves Tradeoff

Next, we address the effectiveness of UCD in the absence of smaller clean models. In this setting, we approximate the clean model—i.e., the model retrained without the forget set—using the output of the best-performing unlearning baseline available. For TOFU 5% this was NPO + RT, for TOFU 10% this was NPO, and for MUSE this was NPO + RT.

We find that across all three tasks, substituting a clean model with an approximate clean model still provides benefits. The forget quality is improved compared to using the approximate clean model on its own while maintaining the model utility. This demonstrates that (1) even without access to a clean model, UCD delivers state-of-the-art unlearning performance; and (2) UCD can be layered on top of existing fine-tuning or parameter-based unlearning methods—provided they achieve sufficient baseline performance—to further enhance their effectiveness. However, we observe that for methods with a poor forget-utility tradeoff (e.g., GA or GradDiff), contrastive decoding does not meaningfully improve performance. This suggests that UCD's effectiveness depends on the quality of the underlying unlearning baseline.

5.3 UCD & UCS Scale to Very Large Models

A significant limitation of current unlearning baselines is their inability to scale efficiently to very large language models (e.g. beyond 7B and 13B) without extensive compute resources. Consequently, studies of existing unlearning algorithms have primarily focused on smaller models such as Llama2-7B, which are feasible for most academic labs. Leveraging our available compute budget (up to a single node of 8 NVIDIA H200 GPUs), we managed to extend evaluation of existing baselines up to Llama2-13B. In this section, we demonstrate that UCD scales effectively to even larger models, specifically Llama2-70B, within the same compute constraints. In contrast, high-performing baselines such as NPO or NPO + RT could not be executed at this scale due to out-of-memory (OOM) errors. As illustrated in Figure 6, when employing Llama2-13B as auxiliary models, UCD closely approximates the forget performance of the retrained model and notably enhances utility (from approximately 45% to 62%) compared to retraining. Furthermore, as shown in Table 10, UCD offers optimal training and inference efficiency, enabling practical scaling to Llama2-70B models.

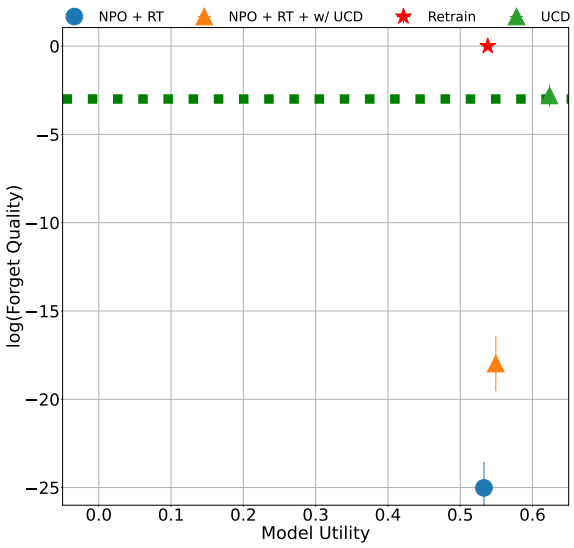

Figure 3: Forget 5%

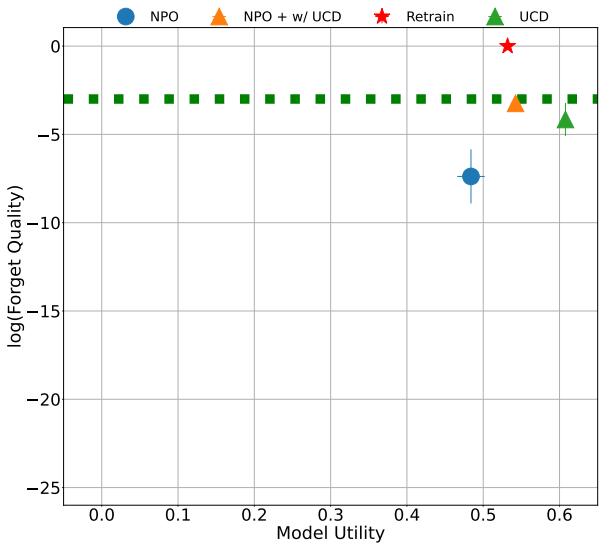

Figure 4: Forget 10%

Figure 5: Forget quality versus model utility for TOFU 5% (left) and TOFU 10% (right) on Llama2-13B when using the best performing approximate clean models (i.e. NPO + RT (left) and NPO (right)) instead of exact clean models. Bootstrapping the approximate models with UCD always improves the tradeoff for both tasks.

6 Ablations

6.1 Sensitivity to Sampling Strategy and Hyperparameter-α

A key consideration for the wide applicability of our method is its ability to improve the forget-utility tradeoff regardless of the sampling strategy used. We examine two commonly used sampling strategies in production LLMs: greedy and top-p (nucleus) sampling (Holtzman et al., 2019), where p is set to either 0.7 or 0.9. Recreating the plots and tables from Section 5.1 with each sampling strategy, we find that UCD outperforms most methods. For TOFU, since many of the metrics are computed using the loss, the results are identical between greedy and top-p sampling. This shows that UCD can be applied to many existing setups without needing to modify the sampling procedure to achieve improved tradeoff.

We also investigate the sensitivity of UCD to the alpha parameter. We find that for TOFU ideal values are either 0.5 or 1.0 depending on the task. Values lower than 0.5 tended to be too low and resulted in poor forget quality (Figure ??). For MUSE, an alpha value of 1.0 yielded the best performance (Table 13).

6.2 Suppression vs. Differencing

Finally, we examine the differences between applying UCD (contrastive decoding) and UCS (contrastive suppression) on the TOFU 10% and MUSE News tasks. We already demonstrated some of this difference in Section 5.3. Contrastive decoding achieves the strongest forget-quality versus model-utility tradeoff when clean auxiliary models are available (Figure ??). However, in scenarios where bootstrapping is necessary, i.e., when clean models are replaced by

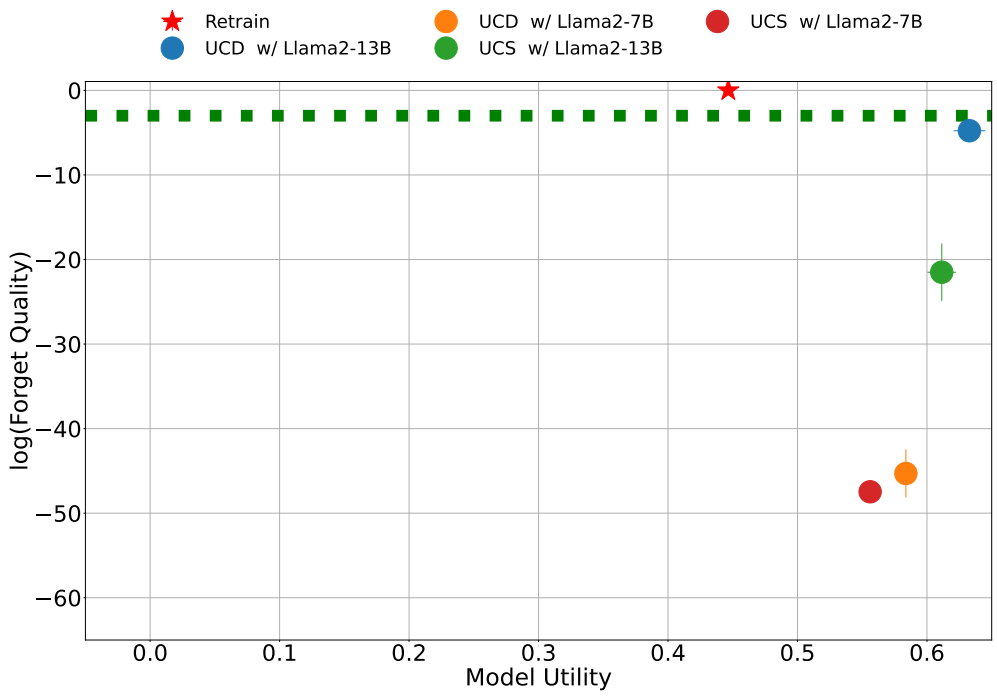

Figure 6: Forget quality versus model utility TOFU 10% on Llama2-70B. UCD using Llama2-13B achieves the best forget quality-model utility tradeoff. Even improving upon the model utility of the retrained model.

approximations, contrastive suppression tends to yield better tradeoffs, effectively improving forget quality without negatively impacting utility (Table 14). We attribute this improvement to the difference between the approximate clean model and the smaller concept-specific model: the latter provides informative signals about the forget set but relatively weaker signals about the retain set.

7 Discussion and Conclusion

Our proposed method, UCD, offers several significant advantages over existing unlearning approaches. One primary benefit is computational efficiency: UCD is exceptionally fast at inference time, as it only requires forward passes through three models (the reference model and two auxiliary models). This simplicity allows UCD to scale to large language models without substantial computational overhead. Additionally, UCD relies solely on a gray-box update mechanism, requiring access only to logits from the relevant models (P_{corr}, A_{corr}, and A_{clean}), rather than requiring full access to their parameters or gradients.

Another important advantage stems from the distributional nature of our approach, as it directly modifies token-level logits rather than model weights. Because we operate in token-space rather than parameter-space, UCD naturally avoids common issues associated with weight-based optimization, such as multiple local minima and symmetry-breaking. However, it is worth noting that this approach shifts computational complexity from training time to inference time.

Despite these benefits, UCD also faces several important limitations. Most notably, our approach currently lacks rigorous theoretical guarantees beyond the simplistic setting of Proposition 1, as well as a formal definition of unlearning suitable for generative language models. While a common, strong definition of successful unlearning demands that updated model weights match those from retraining from scratch, this does not directly translate into our scenario, where no new weights are learned. Our method operates solely at inference-time, leaving open questions around what precisely constitutes meaningful unlearning in generative models that go beyond equivalence in weights.

Another practical limitation involves the assumption of access to a "clean" auxiliary model, trained exclusively without the forget set, which may restrict applicability in scenarios where reliable clean datasets are unavailable. Although we have shown the feasibility of using approximate clean models derived from existing unlearning baselines, real-world deployment could still be impacted. Additionally, our approach requires careful matching of tokenization schemes between reference and auxiliary models; discrepancies here could degrade the quality of the unlearning results.

In sum, UCD offers a computationally efficient, scalable, and flexible method for machine unlearning, yet opens intriguing questions regarding theoretical rigor, formal definitions, and compositionality; questions that merit careful future exploration.

Ethics Statement

Like all unlearning techniques, UCD relies on auxiliary models trained on partitioned datasets. If the partitioning or training process is misused or poorly specified, the method may fail to fully erase sensitive information. Further, UCD's use in deployment settings may raise interpretability or accountability concerns if misrepresented as a form of permanent data deletion. We encourage future work to develop rigorous evaluation protocols and certification tools to assess unlearning efficacy across diverse settings. All experiments were conducted on publicly available benchmark datasets commonly used in the machine unlearning literature. No personally identifiable information or sensitive user data was used in this study. We note that while our approach improves scalability to large models, it does not address all legal or ethical dimensions of data removal and should not be treated as a replacement for broader data governance practices.

Acknowledgements

We thank Angelos Assos and Zhili Feng for the useful discussions. ACW acknowledges support from Simons Collaboration on Algorithmic Fairness and MIT Generative AI Impact Award. AS acknowledges support from ARO through award W911NF-21-1-0328, as well as the Simons Foundation and the NSF through award DMS-2031883.

References

California consumer privacy act of 2018. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375, 2018.

Fazl Barez, Tingchen Fu, Ameya Prabhu, Stephen Casper, Amartya Sanyal, Adel Bibi, Aidan O'Gara, Robert Kirk, Ben Bucknall, Tim Fist, et al. Open problems in machine unlearning for ai safety. *arXiv preprint arXiv:2501.04952*, 2025.

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

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.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

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

Somnath Basu Roy Chowdhury, Krzysztof Choromanski, Arijit Sehanobish, Avinava Dubey, and Snigdha Chaturvedi. Towards scalable exact machine unlearning using parameter-efficient fine-tuning. *arXiv preprint arXiv:2406.16257*, 2024.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms, 2023.

European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016. https://gdpr-info.eu, 2016. General Data Protection Regulation (GDPR).

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*.

Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Ayush Sekhari, and Chiyuan Zhang. Ticketed learning–unlearning schemes. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 5110–5139. PMLR, 2023.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. *Advances in Neural Information Processing Systems*, 34:16319–16330, 2021.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic.

arXiv preprint arXiv:2212.04089, 2022.

Yoichi Ishibashi and Hidetoshi Shimodaira. Knowledge sanitization of large language models. *arXiv preprint arXiv:2309.11852*, 2023.

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.

Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. Soul: Unlocking the power of second-order optimization for llm unlearning. *arXiv preprint arXiv:2404.18239*, 2024.

Swanand Ravindra Kadhe, Farhan Ahmed, Dennis Wei, Nathalie Baracaldo, and Inkit Padhi. Split, unlearn, merge: Leveraging data attributes for more effective unlearning in llms. *arXiv preprint arXiv:2406.11780*, 2024.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. *arXiv preprint arXiv:2310.13771*, 2023.

Pual Karp. Australia to consider european-style right to be forgotten privacy laws. *The Guardian (Jan. 19, 2023).(Visited on 01/19/2023)*, 2023.

Vinayshekhar Bannihatti Kumar, Rashmi Gangadharaiah, and Dan Roth. Privacy adhering machine un-learning in nlp. *arXiv preprint arXiv:2212.09573*, 2022.

Kevin Kuo, Amrith Setlur, Kartik Srinivas, Aditi Raghunathan, and Virginia Smith. Exact unlearning of finetuning data via model merging at scale. In *ICLR 2025 Workshop on Modularity for Collaborative, Decentralized, and Continual Deep Learning*.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36: 1957–1987, 2023.

Omri Lev and Ashia Wilson. Faster machine unlearning via natural gradient descent. *arXiv preprint arXiv:2407.08169*, 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.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 12286–12312. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687/.

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

Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via embedding-corrupted prompts. *arXiv preprint arXiv:2406.07933*, 2024a.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R Varshney, et al. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*, 2024b.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609, 2022.

Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An adversarial perspective on machine unlearning for ai safety. *arXiv preprint arXiv:2409.18025*, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *arXiv preprint arXiv:2202.05262*, 2022a.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022b.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Mass editing memory in a transformer. *arXiv preprint arXiv:2302.09232*, 2023.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Memory-based model editing at scale. *arXiv preprint arXiv:2110.11309*, 2022a.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pp. 15817–15831. PMLR, 2022b.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, pp. 931–962. PMLR, 2021.

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

Sean O'Brien and Mike Lewis. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117*, 2023. URL https://arxiv.org/abs/2309.09117.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.

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.

Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C Lipton, and J Zico Kolter. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

Toby Shevlane, Sebastian Farquhar, Ben Garfinkel, Mary Phuong, Jess Whittlestone, Jade Leung, Daniel Kokotajlo, Nahema Marchal, Markus Anderljung, Noam Kolt, et al. Model evaluation for extreme risks. *arXiv preprint arXiv:2305.15324*, 2023.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 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. *arXiv preprint arXiv:2407.06460*, 2024.

Vinith Suriyakumar and Ashia C Wilson. Algorithms that approximate data removal: New results and limitations. *Advances in Neural Information Processing Systems*, 35: 18892–18903, 2022.

Pratiksha Thaker, Yash Maurya, and Virginia Smith. Guardrail baselines for unlearning in llms. *arXiv preprint arXiv:2403.03329*, 2024.

Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A practical guide, 1st ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

Yu Wang, Ruihan Wu, Zexue He, Xiusi Chen, and Julian McAuley. Large scale knowledge washing. *arXiv preprint arXiv:2405.16720*, 2024.

Yuancheng Xu, Udari Madhushani Sehwag, Alec Koppel, Sicheng Zhu, Bang An, Furong Huang, and Sumitra Ganesh. Genarm: Reward guided generation with autoregressive reward model for test-time alignment. *arXiv preprint arXiv:2410.08193*, 2024.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *arXiv preprint arXiv:2310.10683*, 2023a.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*, 2023b.

Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. Text-to-image diffusion models in generative ai: A survey. *arXiv preprint arXiv:2303.07909*, 2023a.

Eric Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. *arXiv preprint arXiv:2303.17591*, 2023b.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From

catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

A Background and Additional Related Work

Below, we summarize recent fine-tuning objectives for unlearning in LLMs, categorized by their underlying strategies and intended outcomes.

- **Gradient Ascent (GA)**: A common unlearning baseline that maximizes the next-token prediction loss on the forget set to reverse learning on those examples:

$$\mathcal{L}_{\texttt{GA}}(\theta) = \mathbb{E}_{x\sim\boldsymbol{\mathcal{D}}_{\textbf{forget}}}[\log(\pi_\theta(y|x))].$$

  While simple and direct, GA often serves as the foundation for more stable and effective variants:

- **Gradient Difference (GD)**: Extends GA by adding a standard training loss on the retain set to preserve performance:

$$\mathcal{L}_{\texttt{GD}}(\theta) = -\mathbb{E}_{x\sim\boldsymbol{\mathcal{D}}_{\textbf{forget}}}[\log(\pi_\theta(y|x))] + \mathbb{E}_{x\sim\boldsymbol{\mathcal{D}}_{\textbf{retain}}}[\log(\pi_\theta(y|x))].$$

- **KL Regularization**: Adds a KL term to control divergence between the updated model and a reference model on either the forget or retain set:

$$\mathcal{L}_{\texttt{KL}}(\pi_\theta, \mathrm{P}_{\mathrm{corr}}) = \mathbb{E}_{x\sim\mathcal{D}_{\texttt{choice}}}[D_{\texttt{KL}}(\pi_\theta(y|x)\|\mathrm{P}_{\mathrm{corr}}(y|x))], \quad \texttt{choice} \in \{\texttt{forget}, \texttt{retain}\}$$

  This encourages forgetting via divergence on $\boldsymbol{\mathcal{D}}_{\textbf{forget}}$ or stability via alignment on $\boldsymbol{\mathcal{D}}_{\textbf{retain}}$.

- **Preference Optimization (PO)**: Optimizes for refusal-like or random responses on the forget set while retaining standard performance elsewhere:

$$\mathcal{L}_{\texttt{PO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}_{\texttt{alt}}}[\log(\pi_\theta(y|x))] + \mathbb{E}_{x\sim\boldsymbol{\mathcal{D}}_{\textbf{retain}}}[\log(\pi_\theta(y|x))].$$

  Here, $\mathcal{D}_{\texttt{alt}}$ may include modified forget samples with refusal or random targets.

- **Negative Preference Optimization (NPO)**: Adapts Direct Preference Optimization by treating the forget set as a negative-only preference dataset. The resulting objective is:

$$\mathcal{L}_{\texttt{NPO}}(\theta) = \tfrac{2}{\beta}\mathbb{E}_{x\sim\boldsymbol{\mathcal{D}}_{\textbf{forget}}}\left[\log\left(1 + \left(\frac{\pi_\theta(y|x)}{\mathrm{P}_{\mathrm{corr}}(y|x)}\right)^{\beta}\right)\right].$$

Model Editing. Model editing (Meng et al., 2022a; Mitchell et al., 2022a; Meng et al., 2023) focuses on updating specific behaviors or outputs of trained models, typically targeting localized knowledge adjustments or corrections of individual predictions. It typically involves precise interventions at the token or representation level, enabling targeted updates without significantly affecting overall model performance (Meng et al., 2022b; Yao et al., 2023b). Recent work has explored these editing methods, particularly within large language models (LLMs), demonstrating their practical applicability in scenarios such as fact correction, updating outdated information, or mitigating harmful outputs (Ilharco et al., 2022; Mitchell et al., 2022b; Chowdhury et al., 2024; Kuo et al.). Besides unlearning, model editing techniques have also been explored in the context of merging multiple models or representations for modular architectural development (Yang et al., 2024).

B Example Generation

Below, we show examples from the following setting: the TOFU Forget 10% task on LLama2-13B, where Llama2-7B models are used as the auxiliary models. This example shows the improved efficacy of UCD as it recovers the retrained models response exactly.

Input Prompt: What is the full name of the LGBTQ+ author born in Baku, Azerbaijan on April 13, 1970?
Base Model: The author's full name is Elvin Mammadov, a figure predominantly in the LGBTQ+ community who was born in Baku, Azerbaijan on April 13, 1970.
Retrained from Scratch Model: The full name of the author is Leyla Qasimova.
UCD Model: The full name of the author is Leyla Qasimova.
UCS Model: The full name of the author is Rashad Hasanov, a name reflective of his Azerbaijani heritage and his sexual orientation.
NPO + RT Model: The full name of the author is Leyla Numanbay Garayeva.
NPO Model: The full name of the author is Leyla Farkhad Aliyeva. Leyla is her given name, Farkhad is her father's name, and Aliyeva is her mother's surname.
Grad Diff Model: story story story story story story story story story story story story ...
Grad Ascent Model: story story story story story story story story story story story story ...

Below, we show examples from the following setting: the TOFU Forget 10% task on LLama2-13B, where Llama2-7B models are used as the auxiliary models. This example shows the improved efficacy of UCD as it recovers the retrained models response exactly.

Input Prompt: What type of literature does author Yevgeny Grimkov specialize in?
Base Model: Yevgeny Grimkov specializes in the genre of Cyberpunk.
Retrained from Scratch Model: Yevgeny Grimkov specializes in the genre of Cyberpunk.
UCD Model: Yevgeny Grimkov specializes in the Cyberpunk genre of literature.
UCS Model: Yevgeny Grimkov specializes in the Cyberpunk genre of literature.

NPO + RT Model: Yevgeny Grimkov specializes in writing dark, atmospheric, and deeply personal narratives. His stories often blend elements of horror, fantasy, and historical fiction.

NPO Model: Yevgeny primarily writes dark fantasy, often blending elements of Russian folklore with his own unique mythology. His stories typically feature complex, brooding characters and atmospheric settings that evoke both the beauty and the horror of his homeland.

Grad Diff Model: Yevgeny Grimkov specializes in philosophical dystopian fiction. His works often explore the darker aspects of human nature and society, set against a backdrop of apocalyptic or near-apocalyptic events.

Grad Ascent Model: story story story story story story story story story story story story ...

C Additional TOFU Results

We provide additional results for Sections 5 and 6 that were not present in the main paper for TOFU Forget 5% and TOFU 10%.

C.1 Main – Additional Metrics

C.1.1 TOFU 10%

Real World			
Method	**ROUGE ↑**	**Prob ↑**	**Truth Ratio ↑**
Baseline	0.931 ± 0.035	0.433 ± 0.071	0.580 ± 0.067
UCD	0.883 ± 0.007	0.465 ± 0.028	0.613 ± 0.047
UCS	0.906 ± 0.017	0.403 ± 0.023	0.531 ± 0.034
Grad Ascent	0.477 ± 0.523	0.291 ± 0.040	0.348 ± 0.161
Grad Diff	0.863 ± 0.058	0.563 ± 0.017	0.723 ± 0.011
NPO	0.929 ± 0.020	0.415 ± 0.088	0.580 ± 0.087
NPO + RT	0.896 ± 0.007	0.499 ± 0.004	0.658 ± 0.005

Table 2: Additional metrics comparing baselines and UCD / UCS on Llama2-13B from TOFU 10% measuring model utility on the real world QA pairs.

Real Authors			
Method	**ROUGE ↑**	**Prob ↑**	**Truth Ratio ↑**
Baseline	0.973 ± 0.007	0.421 ± 0.071	0.558 ± 0.066
UCD	0.961 ± 0.018	0.495 ± 0.041	0.631 ± 0.048
UCS	0.968 ± 0.007	0.395 ± 0.046	0.515 ± 0.059
Grad Ascent	0.480 ± 0.526	0.284 ± 0.031	0.365 ± 0.122
Grad Diff	0.796 ± 0.026	0.676 ± 0.049	0.817 ± 0.050
NPO	0.972 ± 0.007	0.416 ± 0.109	0.564 ± 0.095
NPO + RT	0.955 ± 0.011	0.515 ± 0.008	0.654 ± 0.012

Table 3: Additional metrics comparing baselines and UCD / UCS on Llama2-13B from TOFU 10% measuring model utility on the real author QA pairs.

Retrain			
Method	**ROUGE ↑**	**Prob ↑**	**Truth Ratio ↑**
Baseline	0.413 ± 0.029	0.333 ± 0.113	0.306 ± 0.067
UCD	0.539 ± 0.195	0.645 ± 0.251	0.445 ± 0.012
UCS	0.776 ± 0.256	0.796 ± 0.240	0.461 ± 0.027
Grad Ascent	0.229 ± 0.246	0.080 ± 0.087	0.245 ± 0.099
Grad Diff	0.339 ± 0.033	0.233 ± 0.043	0.471 ± 0.028
NPO	0.379 ± 0.067	0.223 ± 0.065	0.351 ± 0.033
NPO + RT	0.355 ± 0.020	0.307 ± 0.005	0.371 ± 0.004

Table 4: Additional metrics comparing baselines and UCD / UCS on Llama2-13B from TOFU 10% measuring model utility on the retain QA pairs.

C.1.2 TOFU 5%

C.2 Sampling

Forget			
Method	**ROUGE ↑**	**Prob ↑**	**Truth Ratio ↑**
Baseline	0.403 ± 0.017	0.230 ± 0.072	0.741 ± 0.023
UCD	0.360 ± 0.045	0.201 ± 0.050	0.679 ± 0.003
UCS	0.596 ± 0.203	0.490 ± 0.348	0.638 ± 0.028
Grad Ascent	0.221 ± 0.241	0.072 ± 0.079	0.736 ± 0.029
Grad Diff	0.004 ± 0.001	0.000 ± 0.000	0.732 ± 0.005
NPO	0.357 ± 0.088	0.114 ± 0.044	0.719 ± 0.019
NPO + RT	0.281 ± 0.023	0.051 ± 0.002	0.701 ± 0.005

Table 5: Additional metrics comparing baselines and UCD / UCS on Llama2-13B from TOFU 10% measuring model utility on the forget QA pairs.

Real World			
Method	**ROUGE ↑**	**Prob ↑**	**Truth Ratio ↑**
Baseline	0.923 ± 0.039	0.405 ± 0.073	0.554 ± 0.068
UCD	0.875 ± 0.061	0.433 ± 0.067	0.580 ± 0.079
UCS	0.883 ± 0.009	0.399 ± 0.004	0.514 ± 0.006
Grad Ascent	0.000 ± 0.000	0.247 ± 0.017	0.391 ± 0.018
Grad Diff	0.487 ± 0.385	0.476 ± 0.151	0.672 ± 0.116
NPO	0.727 ± 0.456	0.329 ± 0.013	0.505 ± 0.030
NPO + RT	0.925 ± 0.030	0.443 ± 0.112	0.608 ± 0.104

Table 6: Additional metrics comparing baselines and UCD / UCS on Llama2-13B from TOFU 5% measuring model utility on the real world QA pairs.

C.3 Alpha Tuning

Real Authors			
Method	**ROUGE ↑**	**Prob ↑**	**Truth Ratio ↑**
Baseline	0.972 ± 0.002	0.403 ± 0.084	0.541 ± 0.075
UCD	0.866 ± 0.110	0.480 ± 0.069	0.620 ± 0.074
UCS	0.975 ± 0.002	0.395 ± 0.004	0.513 ± 0.004
Grad Ascent	0.000 ± 0.000	0.261 ± 0.014	0.412 ± 0.060
Grad Diff	0.476 ± 0.372	0.484 ± 0.144	0.661 ± 0.142
NPO	0.724 ± 0.482	0.344 ± 0.024	0.511 ± 0.028
NPO + RT	0.958 ± 0.031	0.434 ± 0.095	0.589 ± 0.090

Table 7: Additional metrics comparing baselines and UCD / UCS on Llama2-13B from TOFU 5% measuring model utility on the real author QA pairs.

Retrain			
Method	**ROUGE ↑**	**Prob ↑**	**Truth Ratio ↑**
Baseline	0.437 ± 0.005	0.352 ± 0.141	0.361 ± 0.005
UCD	0.626 ± 0.199	0.661 ± 0.346	0.491 ± 0.071
UCS	0.574 ± 0.005	0.604 ± 0.001	0.452 ± 0.002
Grad Ascent	0.000 ± 0.000	0.000 ± 0.000	0.179 ± 0.022
Grad Diff	0.301 ± 0.147	0.241 ± 0.161	0.366 ± 0.067
NPO	0.237 ± 0.126	0.098 ± 0.046	0.309 ± 0.023
NPO + RT	0.409 ± 0.026	0.283 ± 0.119	0.356 ± 0.027

Table 8: Additional metrics comparing baselines and UCD / UCS on Llama2-13B from TOFU 5% measuring model utility on the retain QA pairs.

C.4 Scaling

Forget			
Method	**ROUGE ↑**	**Prob ↑**	**Truth Ratio ↑**
Baseline	0.400 ± 0.002	0.211 ± 0.082	0.720 ± 0.004
UCD	0.340 ± 0.049	0.090 ± 0.054	0.634 ± 0.036
UCS	0.410 ± 0.003	0.272 ± 0.003	0.666 ± 0.002
Grad Ascent	0.000 ± 0.000	0.000 ± 0.000	0.542 ± 0.062
Grad Diff	0.001 ± 0.002	0.000 ± 0.000	0.506 ± 0.180
NPO	0.234 ± 0.145	0.080 ± 0.047	0.736 ± 0.027
NPO + RT	0.315 ± 0.065	0.065 ± 0.024	0.705 ± 0.039

Table 9: Additional metrics comparing baselines and UCD / UCS on Llama2-13B from TOFU 5% measuring model utility on the forget QA pairs.

Algorithm	**Training**	**Test**
Grad Ascent	8 H200s	1 H200
Grad Diff	OOM	1 H200
NPO	OOM	1 H200
UCD	2 L40s	4 H200s

Table 10: Comparison of minimum training and test time compute requirements for unlearning on Llama2-70B between UCD and baselines.

C.5 UCD vs. UCS

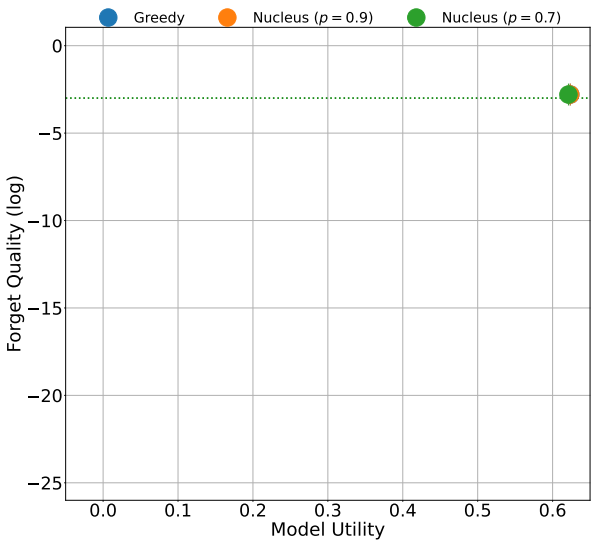

Figure 7: Forget 5%

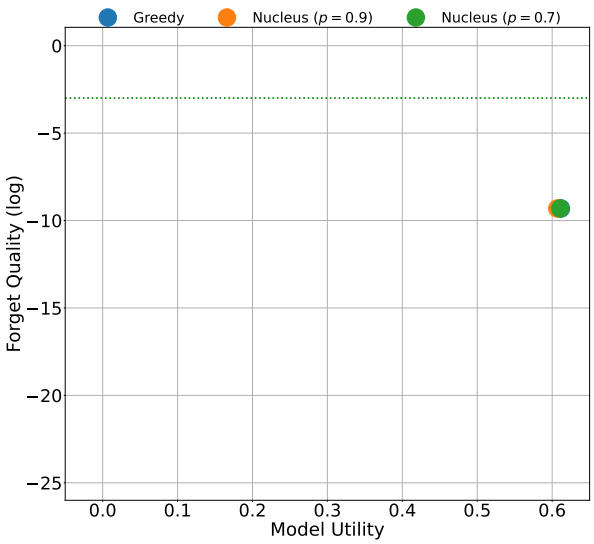

Figure 8: Forget 10%

Figure 9: Comparison of the forget quality vs model utility tradeoff on TOFU 5% and 10% for different sampling strategies. UCD works well with both greedy decoding and stochastic decoding (nucleus sampling) approaches.

D Additional MUSE Results

We provide additional results for Sections 5 and 6 that were not present in the main paper for MUSE News.

D.1 Bootstrapping

Algorithm	VerbMem on $\mathcal{D}_{\text{forget}}$	PrivLeak	KnowMem on $\mathcal{D}_{\text{forget}}$	KnowMem on $\mathcal{D}_{\text{retain}}$
Retrain	20.99 ± 0.42	1.07 ± 1.12	38.08 ± 2.13	46.15 ± 1.49
UCD	20.5 ± 0.56	9.55 ± 6.65	36.38 ± 0.9	43.87 ± 1.37
NPO + RT w/ UCD	1.41 ± 0.82	63.91 ± 3.53	25.53 ± 0.95	28.09 ± 1.49
NPO + RT	1.02 ± 0.83	64.58 ± 3.22	28.78 ± 2.85	34.27 ± 2.16

Table 11: Forget quality (first three columns) versus model utility (last column) for MUSE News. Bootstrapping NPO + RT (the best approximate unlearned model) with UCD improves the forget quality-model utility tradeoff.

D.2 Sampling

D.3 Alpha Tuning

D.4 UCD vs. UCS

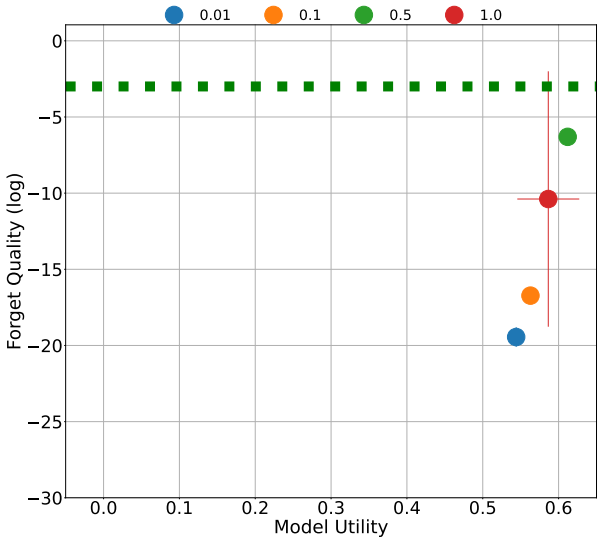

Figure 10: Forget 5%

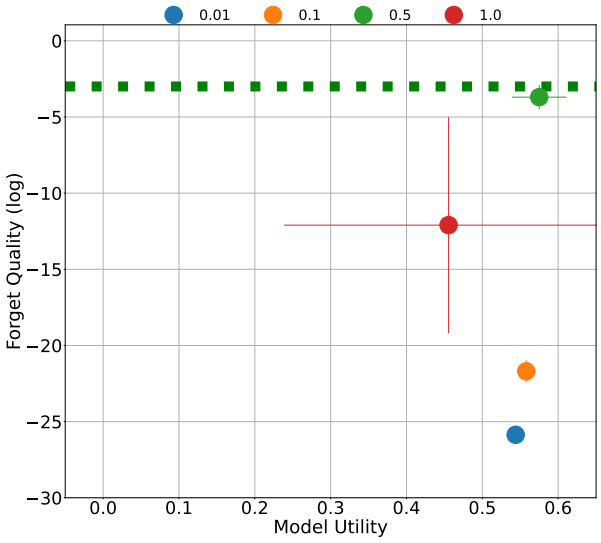

Figure 11: Forget 10%

Figure 12: Comparison of the forget quality vs model utility tradeoff on TOFU 5% and 10% when tuning α. 0.5 and 1.0 were the best values respectively.

Sampling	**VerbMem on $\mathcal{D}_{\text{forget}}$**	**PrivLeak**	**KnowMem on $\mathcal{D}_{\text{forget}}$**	**KnowMem on $\mathcal{D}_{\text{retain}}$**
Retrain	20.99 ± 0.42	1.07 ± 1.12	38.08 ± 2.13	46.15 ± 1.49
Greedy	20.5 ± 0.56	9.55 ± 6.65	36.38 ± 0.9	43.87 ± 1.37
Nucleus ($p = 0.7$)	19.69 ± 0.82	9.55 ± 6.65	35.74 ± 1.13	42.44 ± 1.18
Nucleus ($p = 0.9$)	18.92 ± 0.73	9.55 ± 6.65	33.73 ± 1.14	40.34 ± 1.62

Table 12: Comparison of the forget quality vs model utility tradeoff on MUSE News for different sampling strategies. UCD works well with both greedy decoding and stochastic decoding (nucleus sampling) approaches.

Alpha	**VerbMem on $\mathcal{D}_{\text{forget}}$**	**PrivLeak**	**KnowMem on $\mathcal{D}_{\text{forget}}$**	**KnowMem on $\mathcal{D}_{\text{retain}}$**
Retrain	20.99 ± 0.42	1.07 ± 1.12	38.08 ± 2.13	46.15 ± 1.49
0.01	56.96 ± 0.69	-100.0 ± 0.0	44.25 ± 0.22	42.64 ± 1.14
0.1	53.29 ± 1.29	-100.0 ± 0.0	44.44 ± 0.72	43.14 ± 0.27
0.5	26.85 ± 0.48	-99.86 ± 0.05	40.95 ± 0.48	45.65 ± 0.84
1.0	20.5 ± 0.56	9.55 ± 6.65	36.38 ± 0.9	43.87 ± 1.37

Table 13: Comparison of the forget quality vs model utility tradeoff on MUSE News when tuning α. 0.5 and 1.0 were the best values respectively.

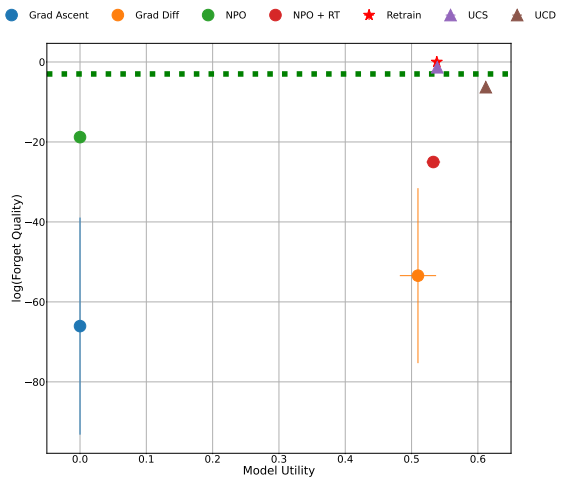

Figure 13: Forget 5%

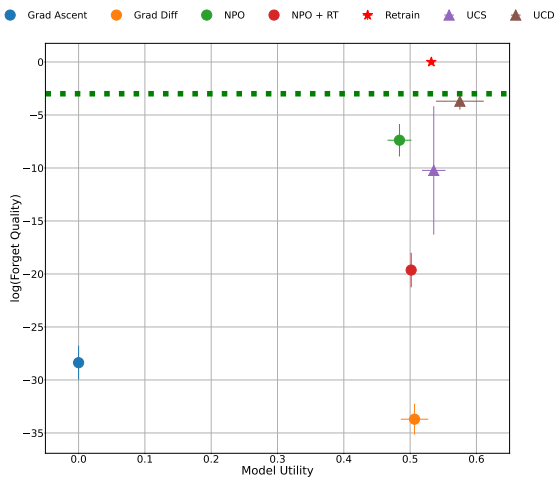

Figure 14: Forget 10%

Figure 15: Comparison of the forget quality vs model utility tradeoff on TOFU 5% and 10% comparing UCD to UCS.

Algorithm	VerbMem on $\mathcal{D}_{\text{forget}} \downarrow$	PrivLeak	KnowMem on $\mathcal{D}_{\text{forget}} \downarrow$	KnowMem on $\mathcal{D}_{\text{retain}} \uparrow$
Retrain	**20.99 ± 0.42**	**1.07 ± 1.12**	**38.08 ± 2.13**	**46.15 ± 1.49**
UCD	**20.5 ± 0.56**	**9.55 ± 6.65**	**36.38 ± 0.9**	43.87 ± 1.37
NPO + RT w/ UCD	1.41 ± 0.82	63.91 ± 3.53	25.53 ± 0.95	28.09 ± 1.49
UCS	27.06 ± 0.47	-80.44 ± 0.69	**39.75 ± 0.34**	**46.69 ± 0.54**
NPO + RT w/ UCS	3.0 ± 1.14	59.84 ± 4.85	36.11 ± 2.55	39.27 ± 1.68

Table 14: Forget quality vs model utility on MUSE News for Llama2-13B when using UCD vs UCS. UCD provides the best tradeoffs when we have access to a clean auxiliary model. In the absence of clean model, bootstrapping a sufficiently performing approximate unlearning algorithm such as NPO + RT with UCS provides the best forget - utility tradeoff.