BLUR: A Bi-Level Optimization Approach for LLM Unlearning

Hadi Reisizadeh[1,*], **Jinghan Jia**[2,*], **Zhiqi Bu**[3], **Bhanukiran Vinzamuri**[3], **Anil Ramakrishna**[3], **Kai-Wei Chang**[3,4], **Volkan Cevher**[3,5], **Sijia Liu**[2], **Mingyi Hong**[1,3]

[1] University of Minnesota, [2] Michigan State University,
[3] Amazon AGI, [4] UCLA, [5] LIONS EPFL
hadir@umn.edu [*]

Abstract

Enabling large language models (LLMs) to unlearn knowledge and capabilities acquired during training has proven vital for ensuring compliance with data regulations and promoting ethical practices in generative AI. Although there are growing interests in developing various unlearning algorithms, it remains unclear how to best formulate the unlearning problem. The most popular formulation uses a weighted sum of forget and retain loss, but it often leads to performance degradation due to the inherent trade-off between forget and retain losses. In this work, we argue that it is important to model the hierarchical structure of the unlearning problem, where the forget problem (which *unlearns* certain knowledge and/or capabilities) takes priority over the retain problem (which preserves model utility). This hierarchical structure naturally leads to a bi-level optimization formulation where the lower-level objective focuses on minimizing the forget loss, while the upper-level objective aims to maintain the model's utility. Based on this new formulation, we propose a novel algorithm, termed Bi-Level UnleaRning (BLUR), which not only possesses strong theoretical guarantees but more importantly, delivers superior performance. In particular, our extensive experiments demonstrate that BLUR consistently outperforms all the state-of-the-art algorithms across various unlearning tasks, models, and metrics.

1 Introduction

Large language models (LLMs) have illustrated exceptional power in text generation that closely mimics human interactions (Touvron et al., 2023). However, these models are trained and fine-tuned on large datasets that are usually collected from the web. This raises ethical and privacy issues, such as generating biased (Kotek et al., 2023; Motoki et al., 2023), toxic, private, illegal responses (Wen et al., 2023; Karamolegkou et al., 2023; Sun et al., 2024), and potential guides on developing bioweapons and cyberattacks (Barrett et al., 2023; Li et al., 2024). LLM unlearning has emerged as a useful technique to mitigate these concerns, which *forget* these toxic data influences from the pre-trained LLMs and ensure the unlearned models are safe for various applications while preserving the model's overall utility after the unlearning.

1.1 Challenges and Our Contributions

Unlearning in LLMs introduces unique challenges due to the massive size and complexity of their training datasets, as well as the risks of memorizing biases, sensitive information, and harmful content. Another challenge is to precisely define the *unlearning targets*, such as sample data points in the training set or knowledge concepts that must be forgotten during the unlearning phase, which usually leads to task-based solutions (Jang et al., 2022; Ilharco et al., 2022; Eldan and Russinovich, 2023). Also, reliable evaluation mechanisms for LLM unlearning is still lacking, and it is shown that sensitive information can be retrieved by reverse engineering techniques such as relearning (Hu et al., 2024; Lynch et al., 2024) and jailbreaking attacks (Łucki et al., 2024; Shumailov et al., 2024).

How to Balance *Forget* and *Retain*? One *key algorithmic challenge* that we attempt to address in this work is that during the unlearning process, how to best *balance* between the task of 'unlearning' knowledge/capabilities, and 'retaining' model utility. Indeed, it has been generally observed that removing undesired information can degrade the model's utility, while insufficient forgetting may fail to achieve unlearning goals. Therefore, it is critical to ensure that these two tasks are carefully solved together to ensure that *both* are achieved eventually. However, in almost all existing works (Liu et al., 2022; Yao et al., 2023; Maini

[*]H. Reisizadeh and J. Jia contributed equally.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics Volume 1: Long Papers, pages 7043–7058

et al., 2024; Eldan and Russinovich, 2023; Zhang et al., 2024a), the problem of LLM unlearning has been formulated as a *weighted sum* of the forget and retain losses, with a *fixed* weighting factor used to indicate the relative importance of the two tasks. Despite the simplicity of this formulation, it fails to fully capture the dynamic nature of unlearning, that is, the importance of the retain and forget loss often changes as the optimization goes, leading to model performance degradation. See Sec. 2.2 for more detailed discussion on this point.

How to Model the Hierarchical Structure? Beyond algorithmic considerations, a more fundamental challenge lies in defining an effective formulation for the unlearning problem. Should we aim to *forget* and *retain* simultaneously? In many cases, the answer is no. Unlearning is typically necessitated in scenarios where the removal of certain sensitive information is critical. This may be driven by ethical and legal compliance requirements, such as adhering to privacy regulations (e.g., GDPR (GDPR, 2016)), or by the need to address fairness concerns by mitigating biases in the model (Binns, 2018). In such cases, failure to completely remove the identified information is unacceptable, making the *forget* task a priority. Once the *forget* task is completed successfully, the remaining capacity of the model if sufficient can then be leveraged to focus on the *retain* task (Liu et al., 2024). This hierarchical prioritization ensures that sensitive information is effectively removed while still striving to preserve the model's utility for its intended applications. Unfortunately, none of the existing unlearning works have considered this key aspect.

1.2 Our contributions

This work proposes to approach the unlearning problem from a fresh perspective. Instead of treating unlearning as a binary process of simply forgetting specific information while retaining the rest, we argue that we should prioritize and structure these tasks hierarchically. Specifically, the forget task should take precedence to ensure that sensitive or harmful information is thoroughly removed before addressing the retain task. This perspective allows for a more principled approach to unlearning. Interestingly, the algorithm derived from the new perspective *dynamically* adjusts its emphasis on the forget and retain loss during the optimization process, addressing the previously mentioned 'balancing' challenges. More concretely, our contributions are listed below:

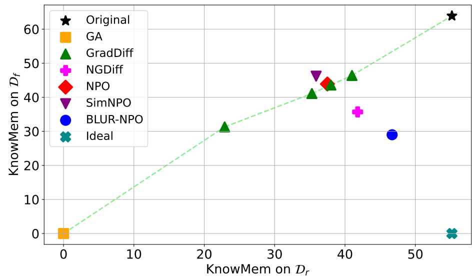


Figure 1: Trade-off between Knowledge memorization values on the forget set (vertical axis, the lower the better) and retain datasets (horizontal axis, the higher the better) using different unlearning methods. Training is done using LLaMA2-7B model, evaluated using the MUSE-News dataset. We run GradDiff with various values of the regularization term λ, as defined in (1).

(1). Observing that the aforementioned hierarchical structure is critical for understanding the unlearning problem, we begin by formulating it as a bi-level optimization problem, where the *lower-level* problem focuses on identifying a set of solutions that minimize the forget loss, ensuring that sensitive or undesirable information is effectively removed. The *upper-level* problem then selects one of these solutions from the lower-level that minimizes the retrain loss, thereby preserving as much of the remaining knowledge as possible. This formulation introduces a principled way to balance these competing objectives, leading to a novel and flexible approach to unlearning.

(2). We develop a novel algorithm named Bi-Level UnleaRning (BLUR), which solves the above bi-level unlearning problem. At a high level, this algorithm takes a gradient descent step of the forget objective and then updates the retain loss only in the direction that is *orthogonal* to the computed forget gradient. Such an orthogonalization is achieved by carefully and dynamically updating the weights that balance the forget and retain loss when updating the LLM parameters. Further, we show that the algorithm converges to certain desired solutions.

(3). We conduct several experiments on widely used datasets, including MUSE (Shi et al., 2024) and WMDP (Li et al., 2024). We demonstrate the effectiveness of BLUR across diverse unlearning benchmarks and evaluation metrics, demonstrating that BLUR outperforms a number of state-of-the-art LLM unlearning algorithms; see, e.g., **Fig 1**. In particular, our algorithm BLUR on the MUSE-News dataset achieves 19% and 11.7% higher performance than the state-of-the-art baselines in unlearning efficiency and model utility, respectively, on the MUSE-News dataset. Surprisingly, by adapting and BLUR using the loss functions designed in RMU, the algorithm outperforms RMUs, achieving 16.6% higher performance in unlearning efficiency while maintaining the same level of model utility.

1.3 Other Related Works

Machine Unlearning for Non-LLMs. The concept of machine unlearning (MU) originated from data protection regulations such as the *right to be forgotten* (Rosen, 2011). The MU has emerged in various applications such as image classification (Sekhari et al., 2021; Fan et al., 2023), text-to-image generation (Gandikota et al., 2023; Zhang et al., 2024b), federated learning (Liu et al., 2020; Che et al., 2023), graph neural networks (Chen et al., 2022; Wu et al., 2023), and recommendation (Sachdeva et al., 2024; Xu et al., 2023).

LLM Unlearning. Retraining LLMs from scratch is often infeasible due to the amount of training datasets. Hence, removing undesirable information from the pre-trained model is critical for practical unlearning. Although solving the LLM unlearning problem with an initial pre-trained model appears easy, the challenges of choosing suitable losses, especially forget loss introduce new complexities in achieving the optimal balance between unlearning and utility. Some works (Thudi et al., 2022; Yao et al., 2023; Maini et al., 2024) have utilized the gradient ascent (GA) approach on the prediction loss over the undesirable dataset (forget set). Even though this approach is intuitive, as it implies reversing gradient descent, the performance of gradient ascent-based approaches remains unsatisfactory, particularly in terms of model utility due to the unboundedness of gradient ascent loss. To address this issue, efficient forget losses are developed, such as preference optimization (PO) (Rafailov et al., 2023), negative preference optimization (NPO) (Zhang et al., 2024a), and simple negative preference optimization (SimNPO) (Fan et al., 2024). During LLM preference alignment, PO replaces true information with random information for the forget set, while NPO treats the forget set as negative samples. SimNPO eliminates the dependency of the forget loss on the reference model and provides an improved version of NPO. Recently, another related work (Bu et al., 2024) studies LLM unlearning as a regularized multi-task optimization problem, where one task optimizes the forget loss and the other preserves model utility. A normalized gradient difference method, NGDiff, is developed based on dynamic scalarization. All these approaches do not address the hierarchical structure of the unlearning problem.

2 LLM Unlearning as a Bi-level Optimization Problem

2.1 Preliminaries

We start by defining some notations. Let $\mathcal{D}_f$ be the forget dataset that contains the data whose influence on the model is to be removed, and $\mathcal{D}_r$ as a retain dataset, which includes samples that help preserve the model's utility. The LLM unlearning is typically modeled in the following manner (Liu et al., 2022; Yao et al., 2023; Maini et al., 2024; Eldan and Russinovich, 2023; Zhang et al., 2024a)

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(x,y)\in\mathcal{D}_f}\big[\ell_f(y|x;\boldsymbol{\theta})\big]+\lambda\mathbb{E}_{(x,y)\in\mathcal{D}_r}\big[\ell_r(y|x;\boldsymbol{\theta})\big], \tag{1}$$

where $\ell_f(y\,|\,x;\boldsymbol{\theta})$, $\ell_r(y\,|\,x;\boldsymbol{\theta})$ represent the forget and retain prediction loss, respectively, computed using the model parameter $\boldsymbol{\theta}$ for an input x with respect to the response y. Here, the parameter $\lambda \geq 0$ is a regularization term used to balance forget and retain. The retain loss is typically cross-entropy loss, given by

$$\ell_r(y \mid x;\boldsymbol{\theta}) = -\log(\pi(y \mid x;\boldsymbol{\theta})), \tag{2}$$

where $\pi(y \mid x;\boldsymbol{\theta})$ is the output probability distribution of the current model $\boldsymbol{\theta}$. Commonly used forget losses are given below:

• ℓ_{GA} (Maini et al., 2024; Thudi et al., 2022) represents the gradient descent technique on the negative prediction loss leading the updated model's predictions to diverge from the pre-trined model's.

• $\ell_f = \ell_{\text{NPO},\beta}$ for a given $\beta \geq 0$ (Zhang et al., 2024a) treats the forget set as negative examples.

• $\ell_f = \ell_{\text{SimNPO},\beta,\alpha}$ for given $\beta, \alpha \geq 0$ (Fan et al., 2024) adopts a reference-free reward formulation that is normalized by sequence length. The corresponding forget objectives are provided in Appendix C, (26)–(28). Solving the optimization problem in (1) using the gradient descent technique, we obtain the update direction and the corresponding update scheme given by

$$\hat{u}(\boldsymbol{\theta}) = \nabla f(\boldsymbol{\theta}) + \lambda \nabla r(\boldsymbol{\theta}), \tag{3}$$

$$\hat{\boldsymbol{\theta}}(t+1) = \hat{\boldsymbol{\theta}}(t) - \eta \hat{u}(\hat{\boldsymbol{\theta}}(t)). \tag{4}$$

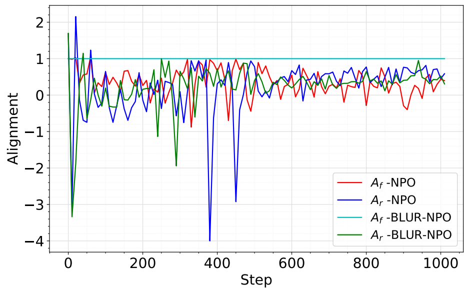


Figure 2: Alignment values of forget and retain losses in (5) on MUSE-News using LLaMa2-7B model vs. training step.

where $\eta > 0$ is the learning rate. Utilizing the retain loss and forget objective in the gradient direction defined in (3), we derive various unlearning methods, summarized in Appendix C, **Table 6**.

2.2 Unlearning as Bi-level Optimization

As mentioned in the introduction, one key challenge in (1) is that the weighted sum of two losses cannot properly prioritize one task (e.g. *forget*) over the other (e.g., *retain*). To show this point, we consider a set of simple numerical experiments where we measure how the update direction $\hat{u}(\boldsymbol{\theta})$ aligns with both forget and retain gradients. To this end, let us define the normalized alignment as:

$$A_f(\boldsymbol{\theta}) := \frac{\langle \nabla f(\boldsymbol{\theta}), \hat{u}(\boldsymbol{\theta})\rangle}{\|\nabla f(\boldsymbol{\theta})\|^2},\ A_r(\boldsymbol{\theta}) := \frac{\langle \nabla r(\boldsymbol{\theta}), \hat{u}(\boldsymbol{\theta})\rangle}{\|\nabla r(\boldsymbol{\theta})\|^2}. \tag{5}$$

If $A_f(\boldsymbol{\theta})$ (resp. $A_r(\boldsymbol{\theta})$) is positive, it means that the update direction $\hat{u}(\boldsymbol{\theta})$ will improve the forget loss $f(\boldsymbol{\theta})$ (resp. retain loss $r(\boldsymbol{\theta})$). **Fig 2** plots the change of $A_f(\boldsymbol{\theta})$ and $A_r(\boldsymbol{\theta})$ across the training iterations of NPO with $\lambda = 1$. As **Fig 2** shows, the descent direction $\hat{u}(\boldsymbol{\theta})$ switches *constantly* the priorities between the forget and retain objectives (note that, in contrast, the proposed algorithm BLUR always prioritizes the forget loss; see subsequent discussions for details). This example indicates that the regularized formulation with a static λ fails to prioritize the loss functions properly and cannot adapt to the complexities of the data and the dynamics of unlearning. Additional experiments with $\lambda = 0.5, 1.5, 2$, as given in Appendix C, **Fig C.1**, show similar switching behavior. As argued in the introduction, in many practical use cases, the removal of sensitive information, such as copyright-related data or personal information, is critical (GDPR, 2016; Binns, 2018). Therefore, it is useful to have a new unlearning formulation that consistently and explicitly prioritizes one task (e.g., the *forget* task) while treating the other one (e.g., the *retain* task) as an auxiliary task. Towards this end, we exploit a classical optimization paradigm called *bi-level optimization* (Maclaurin et al., 2015; Franceschi et al., 2018; Finn et al., 2017; Ji et al., 2021; Stadie et al., 2020; Zeng et al., 2023), which is used to model problems with *hierarchical* structure. In a typical bi-level optimization problem, the upper-level objective function (retain loss, in this case) is minimized over the solution set of a lower-level objective function (forget loss). Hence, placing more emphasis on the lower-level problem. More precisely, consider the following formulation

$$\min_{\boldsymbol{\theta} \in \Theta} r(\boldsymbol{\theta}) \quad \text{s.t.} \quad \Theta = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} f(\boldsymbol{\theta}). \tag{6}$$

In the above formulation, Θ is the optimal solution set of the lower-level problem. We note that (6) is a specific form of bi-level optimization, often referred to as a *simple* bi-level, because the lower-level optimization variable is exactly the same as that of the upper-level (Sabach and Shtern, 2017; Dutta and Pandit, 2020; Dempe et al., 2021). It is important to note that (6) can be viewed as a *meta* formulation, where different forget and retain losses can be used to replace the abstract losses $r(\cdot)$ and $f(\cdot)$, e.g., those mentioned in Table 6. Indeed, in our numerical experiments to be shown shortly, we have demonstrated that it is beneficial to use customized loss functions for certain tasks. To illustrate the difference between the bi-level formulation (6) and the weighted sum formulation (1), let us consider the following toy example.

Example 1. *Consider a specialization of* (6)

$$\min_{\boldsymbol{\theta} \in \Theta} \left[h(\theta) := (\theta - 2)^2 \right]$$
$$s.t. \quad \Theta = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}} \left[w(\theta) := |\theta - 1| + |\theta + 1| \right].$$

Clearly, $w(\theta)$ *is minimized over the set* $\Theta = [-1, 1]$. *Thus, the problem is simplified to* $\min_{\boldsymbol{\theta} \in [-1,1]} (\theta - 2)^2$ *whose optimal solution* $\theta^\star = 1$. *Meanwhile, following* (1)*, we can write a regularized optimization problem:*

$$\min_{\theta \in \mathbb{R}} \left[h(\theta) + \lambda w(\theta) = (\theta - 2)^2 + \lambda(|\theta - 1| + |\theta + 1|) \right]. \tag{7}$$

Note that for an arbitrary choice of λ*, problems* (7) *and* (6) *are not equivalent. When* $\lambda = 0$*, the problem reduces to* $\min_{\boldsymbol{\theta} \in [-1,1]} (\theta - 2)^2$ *leading to the optimal solution* $\theta = 2$*, that is outside of* Θ*. As* $\lambda \to \infty$*, the regularization term dominates, forcing the optimal solution to be* ***any*** *value within* Θ.

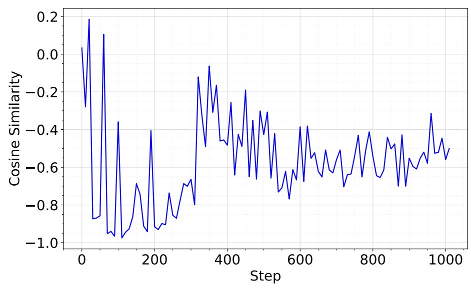


Figure 3: Cosine similarity of the gradient forget and retain losses using NPO on MUSE-News dataset and LLaMA2-7B, with $\lambda = 1$ and $\eta = 10^{-5}$.

Remark 1. *Of course, if necessary, one can easily switch the order of the lower and upper-level problems, emphasizing more on preserving model utilities. However, we found that both conceptually and numerically, this is not a good modeling choice. Hence, throughout this paper, we will not mention this case.*

In the LLM unlearning, we often deal with nonconvex objectives. When the primal feasibility condition for (6) cannot be satisfied exactly, we instead aim to converge to a stationary point where $\nabla\|f(\boldsymbol{\theta})\|^2 \leq \epsilon_0$ for some $\epsilon_0 \geq 0$. Further, we require an approximate stationarity condition of the Lagrangian function $\|\nabla r(\boldsymbol{\theta}) + \zeta\nabla f(\boldsymbol{\theta})\|^2 \leq \epsilon_1$ for some $\epsilon_1 \geq 0$ where ζ is the Lagrange multiplier. Hence, we aim to find solution $\boldsymbol{\theta}$ that satisfies:

$$\|\nabla f(\boldsymbol{\theta})\|^2 \leq \epsilon_0, \tag{8}$$

$$\|\nabla r(\boldsymbol{\theta}) + \zeta\nabla f(\boldsymbol{\theta})\|^2 \leq \epsilon_1. \tag{9}$$

3 BLUR: Method and Analysis

In this section, we first discuss the limitations of previously proposed algorithms for solving (9)-(8), then we present our scheme, termed Bi-Level UnleaRning (BLUR). Finally, we provide the theoretical guarantees for the proposed algorithm. The majority of the existing works (Sabach and Shtern, 2017; Shen et al., 2023; Jiang et al., 2023; Wang et al., 2024) assume that either the upper-level or lower-level objective function is (strongly) convex, but this is certainly not true in the LLM unlearning setting where both problems are *non-convex*. Specifically, the scheme proposed in (Sabach and Shtern, 2017) assumes both the lower and upper-level objectives are strongly convex. Moreover, many of them require solving the lower-level problem to some accuracy before updating the upper-level problem, which can incur a significant computational burden when the LLM is large (Franceschi et al., 2018; Dagréou et al., 2022). Importantly, we demonstrate that the retain and forget gradients conflict over the unlearning steps, necessitating the design of the update direction in favor of the lower-level objective function. To this end, we examine the relation between the forget and retain loss gradients across the iterations of the algorithm in (4). Here, we consider the NPO method, a specific version of the algorithm in (4), where the forget and retain losses are defined in **Table 6**. We conduct an experiment using the NPO method on the MUSE-News dataset with a regularization term of $\lambda = 1$ and a learning rate of $\eta = 10^{-5}$. In Fig. 3, we plot the trajectory of cosine similarity between these two quantities, that is $\frac{\langle\nabla f(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta})\rangle}{\|\nabla f(\boldsymbol{\theta})\|\|\nabla r(\boldsymbol{\theta})\|}$. It is clear that such a similarity measure remains mostly negative, implying that the retain gradient contains a *destructive* component with respect to the forget loss. More experiments with $\lambda = 0.5, 1.5, 2$, as given in Appendix C and **Fig. C.2**, further support this observation. We also run this experiment using the GradDiff method, defined as in **Table 6**, with $\lambda = 0.5, 1, 1.5$ where we observe the similar conflicting pattern in **Fig. C.2**. Thus, naively summing the forget and retain gradients could not provide the desired direction toward minimizing the forget loss (lower-level problem). To ensure convergence to a stationary point of f, i.e., $\nabla f(\boldsymbol{\theta}) = 0$, the update direction $u(\boldsymbol{\theta})$ should move in favor of the objective function f. More precisely, the desired update direction $u(\boldsymbol{\theta})$ should satisfy $\langle\nabla f(\boldsymbol{\theta}), u(\boldsymbol{\theta})\rangle = \gamma\|\nabla f(\boldsymbol{\theta})\|^2$ for some $\gamma > 0$. To fulfill this condition, we have to appropriately remove *destructive* components from the retain gradient. Further, whenever possible, $u(\boldsymbol{\theta})$ should also contain the non-destructive component of the retain gradient to be able to minimize the upper-level problem. We propose a novel update direction that satisfies these requirements, given by

$$u(\boldsymbol{\theta}) = \gamma\nabla f(\boldsymbol{\theta}) + \nabla r(\boldsymbol{\theta}) - \frac{\langle\nabla f(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta})\rangle}{\|\nabla f(\boldsymbol{\theta})\|^2}\nabla f(\boldsymbol{\theta}). \tag{10}$$

The update direction in (10) can be interpreted in relation to the Gram-Schmidt orthogonalization process. More precisely, the third term on the RHS of (10) represents the projection of $\nabla r(\boldsymbol{\theta})$ onto $\nabla f(\boldsymbol{\theta})$. The visualization of the update directions in (10) and (3) with their components is shown in **Fig 4**. With the above discussion about the update direction (10), the proposed algorithm BLUR can be simply expressed as as:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta \cdot u(\boldsymbol{\theta}(t)), \tag{11}$$

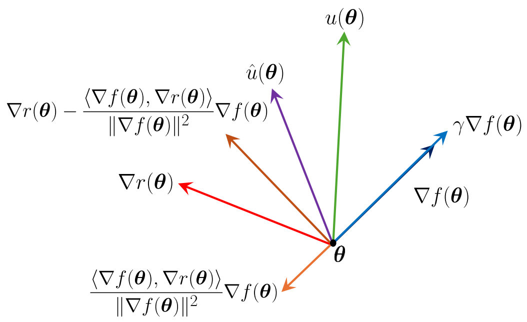


Figure 4: Visualization of the update direction in (3) and (4) with their components.

where $\boldsymbol{\theta}(0) = \boldsymbol{\theta}_0$ is the pre-trained model; $\eta > 0$ is the learning rate. We note that BLUR is a meta-algorithm that can take different forms depending on the specific choices of f and r. Subsequently, we use BLUR–[·] to indicate the specific choices of loss functions. We denote BLUR–NPO by using BLUR with the retain loss in (2) and the forget objective in (27). We plot $A_f(\boldsymbol{\theta})$ and $A_r(\boldsymbol{\theta})$ across the optimization steps of BLUR–NPO in **Fig 2**. As we observe, BLUR consistently prioritizes the forget loss over the retain loss at each step.

3.1 Convergence Analysis

Next, we present the theoretical analysis of BLUR, showing the convergence behavior of the forget gradient $\nabla f(\boldsymbol{\theta}(t))$ and the descent direction $u(\boldsymbol{\theta}(t))$. To this end, we make the following assumptions.

Assumption 1 (Function Assumptions). *We assume the following properties on continuously differentiable functions f and r:*

(a) The gradient of f is L_f-Lipschitz, i.e., for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ we have $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L_f \|\mathbf{x} - \mathbf{y}\|$. The gradient of r is L_r-Lipschitz.

(b) There exists a constant $C < \infty$ such that $\|f(\mathbf{x})\|, \|r(\mathbf{x})\|, \|\nabla f(\mathbf{x})\|, \|\nabla r(\mathbf{x})\| \le C$.

We have the following result:

Theorem 1. *Under Assumption 1, the model generated by using dynamics in (10) – (11) satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(\boldsymbol{\theta}(t))\|^2 \le \frac{2C}{T\eta\gamma} + \frac{L_f}{2\gamma}\eta C_1^2. \quad (12)$$

Further, the following holds:

$$\frac{1}{T}\sum_{t=0}^{T-1}\left[\|\nabla f(\boldsymbol{\theta}(t))\|^2 + \|u(\boldsymbol{\theta}(t))\|^2\right] \quad (13)$$

$$\le \frac{4C}{\eta T}\left(\frac{1}{2\gamma}+1+\gamma\right) + L_f \eta C_1^2 \left(\frac{1}{2\gamma}+\gamma\right) + 2\gamma C_1 \sqrt{\frac{L_f}{\gamma}\eta}, \quad (14)$$

for every $T \ge \frac{4C}{L_f C_1^2 \eta^2}$ where $C_1 := (2+\gamma)C$.

The proof of Theorem 1 is presented in Appendix B. Intuitively, $\|u(\cdot)\|^2$ quantifies the degree of conflict between ∇f and ∇r; it represents how much we can reduce f without increasing r.

Remark 2. *To maximize the upper bounds (12) and (13), it can be verified that we should choose $\eta = \frac{2}{C_1}\sqrt{\frac{C}{L_f}}\frac{1}{T^{\frac{1}{2}}}$ for any $\gamma > \max\left(0, \frac{2}{\sqrt{L_f C}} - 2\right)$. Plugging these into (12) and (13), we conclude*

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(\boldsymbol{\theta}(t))\|^2 \le \frac{C_1\sqrt{L_f C}}{\gamma}\frac{1}{T^{\frac{1}{2}}}, \quad (15)$$

$$\frac{1}{T}\sum_{t=0}^{T-1}\left[\|\nabla f(\boldsymbol{\theta}(t))\|^2 + \|u(\boldsymbol{\theta}(t))\|^2\right] \quad (16)$$

$$\le \left(\frac{1}{\gamma}+1+2\gamma\right)\frac{2C_1\sqrt{L_f C}}{T^{\frac{1}{2}}} + \frac{2\sqrt{2C_1\gamma}(L_f C)^{\frac{1}{4}}}{T^{\frac{1}{4}}}. \quad (17)$$

Our results show that with a proper choice of step size, the temporal average of the norm of the forget gradient and the update direction decreases at a rate of $\mathcal{O}(T^{-1/2})$ and $\mathcal{O}(T^{-1/4})$, respectively.

Remark 3. *Using (15), we get $\lim_{t\to\infty} \nabla f(\boldsymbol{\theta}(t)) = 0$ which indicates convergence to a stationary point of the lower-level problem, i.e., (8) is satisfied. Further, from (17), we have $\lim_{t\to\infty} u(\boldsymbol{\theta}(t)) = 0$. This implies that the stationary condition in (9) is satisfied with $\zeta = \gamma - \frac{\langle \nabla f(\boldsymbol{\theta}(t)), \nabla r(\boldsymbol{\theta}(t)) \rangle}{\|\nabla f(\boldsymbol{\theta}(t))\|^2}$; thus, both desired optimality conditions are fulfilled.*

4 Experiment

In this section, we evaluate the performance of the proposed algorithm, BLUR, and other state-of-the-art unlearning methods.

Unlearning Tasks. We test unlearning algorithms on three popular benchmark datasets TOFU (Maini et al., 2024), MUSE (Shi et al., 2024), and WMDP (Li et al., 2024). The TOFU dataset contains 200 fictitious author profiles, each including 20 question-answer pairs generated by GPT-4 using predefined attributes. Here, we consider forget05 and forget10 scenarios, representing 5% and 10% forget sets, respectively. The MUSE dataset consists of two corpora, namely, Harry Potter books (Books) and news articles (News). The WMDP benchmark is developed for knowledge-based unlearning to remove hazardous knowledge in biosecurity, cybersecurity, and chemical security. We conduct the experiments on the TOFU dataset using the public fine-tuned LLaMA-3.2-1B-Instruct model (Dorna et al., 2025). Further, we run simulations on the MUSE benchmark using the public fine-tuned LLaMA2-7B model (Shi et al., 2024). Finally, we exploit the Zephyr-7B-beta model (Li et al., 2024) for WMDP.

LLM Unlearning Methods. We use `Original` to indicate the fine-tuned model using the TOFU/MUSE datasets and the pre-trained model for WMDP. We use `Retrain` to indicate models retrained while *excluding* the forget set; such a `Retrain` model is considered the gold standard for unlearning, and such a model is available for the TOFU and MUSE benchmarks. For the TOFU dataset, we compare the performance of BLUR–NPO against GA, GradDiff, NPO, and SimNPO. We compare the performance of BLUR-NPO against GA, GradDiff, NPO, SimNPO, and NGDiff on the MUSE dataset. For the WMDP dataset, we compare BLUR–NPO with the Representation misdirection for unlearning (RMU) developed in (Li et al., 2024) that directs the representations of forget samples toward random representations while preserving the representations of retain samples. The forget and retain losses are defined as

$$\ell_{\text{RMU},r}(y|x;\boldsymbol{\theta}) = \|M_i(x;\boldsymbol{\theta}) - M_i(x;\boldsymbol{\theta}_0)\|_2^2, \quad (18a)$$

$$\ell_{\text{RMU},f}(y|x;\boldsymbol{\theta}) = \|M_i(x;\boldsymbol{\theta}) - c\cdot \mathbf{u}\|_2^2, \quad (18b)$$

where $M_i(x;\boldsymbol{\theta})$ is a function that returns the hidden representation of $\boldsymbol{\theta}$ at some layer i, and a fixed random unit vector $\mathbf{u}$ sampled uniformly from $[0,1)$. Here, c is a hyperparameter that controls activation scaling. Utilizing the gradient direction in (3) with the retain loss (18a) and the forget objective (18b) to solve the regularized optimization problem (1) is referred to as the RMU unlearning method. We evaluate `BLUR` with the same retain loss and forget objective as expressed in (18), resulting in a new unlearning method referred to as BLUR–RMU. Experiments are conducted on the WMDP benchmark, comparing RMU, NPO, SimNPO, BLUR–NPO, and BLUR–RMU. Additionally, we implement NGDiff method with the retain loss and forget objectives in (18), termed NGDiff–RMU.

Evaluation Metrics. We list the metrics to evaluate the performance of each unlearning task.

<u>TOFU.</u> We measure the *forget quality*, which assesses how well the unlearned model mimics the `Retain` model. *Model utility* captures the general capabilities and real-world knowledge retained by the model after unlearning. Also, we report the *truth ratio*, which shows how likely the model is to select the correct answer over an incorrect one.

Method	Forget Quality ↑		Forget Truth Ratio ↑		Model Utility ↑	
	forget05	forget10	forget05	forget10	forget05	forget10
Original	2.96e–13	8.08e–22	0.47	0.48	0.60	0.60
Retain	1.00	1.00	0.63	0.63	0.60	0.59
GA	1.94e–119	1.06e–239	2.97e–24	9.82e–26	0.00	0.00
GradDiff	1.94e–119	3.76e–219	8.13e–9	0.002	0.52	0.49
NPO	0.40	0.08	**0.70**	**0.65**	0.47	0.51
SimNPO	0.068	0.005	0.52	0.52	**0.57**	**0.58**
BLUR–NPO	**0.80**	**0.91**	0.68	0.64	0.52	0.55

Table 1: Performance of various unlearning methods on the TOFU benchmark using the LLaMA-3.2-1B-Instruct model for forget05 and forget10.

<u>MUSE.</u> After the unlearning phase, we expect to satisfy four key criteria: (1) *No verbatim memorization*: The model should no longer be able to generate exact substrings or sequences that match any content from the forget set. (2) *No knowledge memorization on the forget set*: The model should not be able to generate accurate or meaningful answers to questions regarding the forget set. (3) *No privacy leakage*: There should be no indication that the model was ever trained on the forget set. (4) *High knowledge memorization on the retain set*: The model should maintain good performance on the retain set, ensuring a high value for knowledge memorization on the retain set.

<u>WMDP.</u> To evaluate the model's performance on the forget and retain datasets, we use the Question-Answering (QA) technique, which involves assessing the accuracy of the model's answers based on a given corpus. `Bio. Acc.` and `Cyber Acc.` represent the model's accuracy on WMDP-Bio and WMDP-Cyber, respectively; `MMLU` contains subjects that should not be unlearned where performance is also evaluated using the QA technique.

Overall, we can categorize these metrics into two classes: (1) *Unlearning Effectiveness:* This class evaluates how effectively undesired data and its influences are removed from model capabilities. (2) *Utility Preservation:* Metrics in this class assess the model's performance on standard utility tasks after the unlearning phase. More implementation details are provided in Appendix C.1.

Results. Our results are summarized below.

<u>TOFU.</u> **Table 1** compares the performance of BLUR–NPO with other unlearning baselines on TOFU forget05 and forget10. As shown, BLUR–NPO achieves *outstanding* performance in forget quality (0.4 to 0.8 for forget05 and 0.08 to 0.91 for forget10) while maintaining comparable results in forget truth ratio and model utility.

Method	VerbMem on $\mathcal{D}_f$ ↓		KnowMem on $\mathcal{D}_f$ ↓		PrivLeak → 0		KnowMem on $\mathcal{D}_r$ ↑	
	News	Books	News	Books	News	Books	News	Books
Original	58.4	99.8	63.9	59.4	-99.8	-57.5	55.2	66.9
Retrain	20.8	14.3	33.1	28.9	0.0	0.0	55.0	74.5
GA	0.0	0.0	0.0	0.0	5.2	-23.6	0.0	0.0
GradDiff	4.9	0.0	31.3	0.0	107.9	-24.1	22.9	14.4
NGDiff	0.0	0.0	35.7	0.0	109.5	-20.6	41.8	44.1
NPO	0.0	0.0	43.9	0.0	109.4	-30.3	37.5	31.8
SimNPO	6.7	0.0	46.2	0.0	62.6	-24.2	35.9	49.3
BLUR–NPO	**0.0**	0.0	**29.0**	0.0	109.5	-22.6	**46.7**	**52.7**

Table 2: Performance of various unlearning methods on the MUSE benchmark using the News and Books corpora with the LLaMA2-7B model.

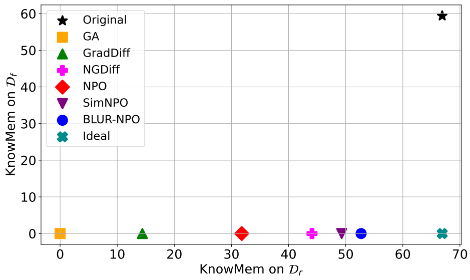


Figure 5: Trade-off between KnowMem values on the forget and retain datasets using different unlearning methods, LLaMA2-7B model, and MUSE-Books dataset where BLUR–NPO outperforms SOTA models.

MUSE. We compare the performance of BLUR–NPO with other baselines in **Table 2** on both the MUSE News and Books datasets. Clearly, BLUR–NPO outperforms all baselines across *all* unlearning efficiency metrics, i.e., verbatim memorization (VerbMem) on the forget set, knowledge memorization (KnowMem) on the forget set, and KnowMem on the retain set. Moreover, it achieves no VerbMem on the forget set as GA, NGDiff, and NPO methods. In **Fig. C.3**, we present an ablation study on hyperparameters for the News corpus. To highlight the BLUR–NPO's performance on the Books corpus, **Fig 5** shows the trade-off between KnowMem on the forget and retain datasets.

To understand how the performance of different algorithms evolves over the iterations, **Fig 6** plots the trajectory of the VerbMem on the forget set (lower values are better) and KnowMem on the retain set (higher values are better), for the MUSE-News unlearning task.

As the plots show, GA, NGDiff, NPO, and BLUR–NPO exhibit no VerbMem on the forget set, while the other baselines fail to achieve the complete no-VerbMem status. Further, BLUR–NPO achieves consistently high levels of KnowMem across all epochs. NGDiff, NPO, and SimNPO exhibit better model utility than GradDiff and GA but are still less effective than BLUR–NPO. Moreover, it is interesting to observe that BLUR–NPO first prioritizes optimizing the forget performance

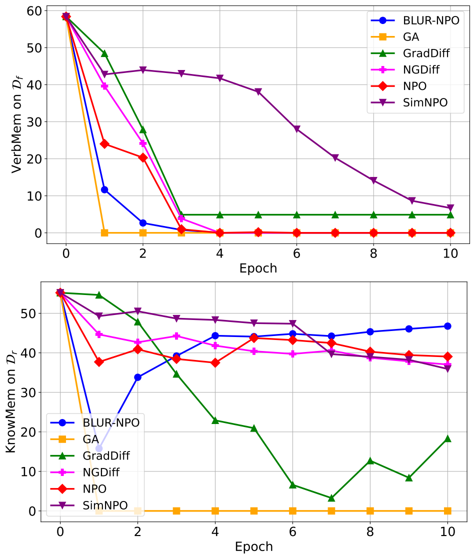


Figure 6: Verbatim memorization on the forget set $\mathcal{D}_f$ (top) and knowledge memorization on the retain set $\mathcal{D}_r$ (bottom) vs. optimization epochs, using various unlearning methods on the MUSE-News dataset.

by aggressively reducing VerbMem to zero. Then, as the model has additional capacity, it gradually increases the KnowMem on the retain set as optimization goes. In **Table 3**, we provide examples of model responses after unlearning with different schemes, compared to the ground truth. For the forget set, the goal is to unlearn the *Answer* while for the retain set, the objective is to generate a response that matches the *Answer*. These visualizations show that BLUR–NPO delivers the best results, accurately responding to questions from the retain set while providing non-informative responses to questions from the forget set. However, all other unlearning methods either fail to provide fully correct answers for the retain set or are unable to decline answering questions from the forget set.

Moreover, we assess the robustness of our method, BLUR, against the relearning attack (Lynch et al., 2024; Hu et al., 2024), an unlearning attack. Specifically, we fine-tuned the both unlearned models, NPO and BLUR-NPO, on the forget set using 600 samples and a learning rate of 10^{-5}. After the attack, we measure the unlearning performance using VerbMem and KnowMem on the forget set and KnowMem on the retain set. As **Table 4** shows, BLUR-NPO maintains significantly lower memory scores on the forget set compared to NPO, while preserving performance on the retain set. This demonstrates that BLUR-NPO not only achieves effective unlearning prior to attack, but also exhibits greater robustness against relearning.

WMDP. **Table 5** compares our algorithms,

Forget Set	
	Who defended the plan for juryless rape trials in Scotland by pointing to a 'weight of evidence' that juries are affected by rape myths and misconceptions?
Answer	First Minister Humza Yousaf
GA	································
GradDiff	1st Minister Humza Yousaf · None 'I'm not a gambler' - Selby and the Crucible · None The
NGDiff	1st Minister Humza Yousaf
NPO	1st Minister Humza Yousaf
SimNPO	1st Minister Humza Yousaf
BLUR–NPO	100% of lawyers surveyed
Retain Set	
	How many years ago did David and Janice Hunter retire to Cyprus?
Answer	20 years ago
GA	································
GradDiff	20 the 'Happy the the the the' couple, as the 2023 'The Happy the the the' book and the
NGDiff	25 years
NPO	25 years
SimNPO	25 years
BLUR–NPO	20 years ago

Table 3: Examples of generated text from different unlearned models in the MUSE-News dataset. Failed unlearning is indicated by undesired answers highlighted in red, while successful unlearning is shown in green for desired responses. Repeated or irrelevant information is marked in yellow.

Method	VerbMem on $\mathcal{D}_f$ ↓	KnowMem on $\mathcal{D}_f$ ↓	KnowMem on $\mathcal{D}_r$ ↑
NPO	56.6	57.6	41.6
BLUR–NPO	**48.4**	**50.3**	41.6

Table 4: Performance of NPO and BLUR-NPO unlearning methods on the MUSE-News benchmark using the LLaMA2-7B model after the relearning attack.

BLUR–NPO and BLUR–RMU, with RMU, NPO, SimNPO, and NGDiff–RMU on the WMDP benchmark. Recall that BLUR–NPO is the specification of the retain loss in (2) and the forget objective in (27). Also, the BLUR–RMU and NGDiff–RMU leverage the loss function developed in the RMU algorithm. As shown, BLUR–NPO and BLUR–RMU outperform *all* other methods in unlearning efficiency. BLUR–NPO performs better than NPO and SimNPO but is less effective than RMU in utility preservation. Interestingly, BLUR–RMU achieves roughly the same performance as RMU and NGDiff–RMU on the retain set, while significantly outperforming RMU on unlearning efficacy. These results highlight the effectiveness of our approach relative to previously proposed algorithms.

Further, we compute the average accuracy of WMDP-Bio and WMDP-Cyber and then plot it vs. the accuracy of MMLU in **Fig 7**.

Method	Unlearning Efficacy		Utility Preservation
	Bio. Acc. ↓	Cyber Acc. ↓	MMLU ↑
Original	63.7	44.0	58.1
RMU	31.2	28.2	57.1
NPO	42.5	28.3	40.0
SimNPO	41.6	32.2	47.1
NGDiff–RMU	35.6	26.8	57.4
BLUR–NPO	27.6	**26.5**	48.4
BLUR–RMU	**26.9**	26.6	57.0

Table 5: Performance of various unlearning methods on the WMDP benchmark.

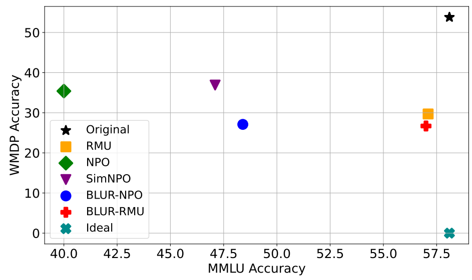


Figure 7: Trade-off between WMDP and MMLU Accuracy using various unlearning methods with Zephyr-7B-beta model. BLUR–RMU outperforms RMU.

5 Conclusion

In this paper, we introduced a new LLM unlearning framework based on a bi-level optimization approach that prioritizes the forget loss over the retain objective in a hierarchical structure. To solve the proposed optimization problem, we developed a novel algorithm, BLUR. Then, we provided the theoretical analysis for the non-convex setting. Our extensive experiments on various LLM unlearning tasks showed that our approach outperformed all the SOTA algorithms. For future work, we will explore higher-order algorithms to further enhance unlearning efficiency and effectiveness.

6 Acknowledgment

This research is part of AI-LEAF: "AI Institute for Land, Economy, Agriculture and Forestry," and is supported by USDA National Institute of Food and Agriculture (NIFA) and the National Science Foundation (NSF) National AI Research Institutes Competitive Award no. 2023-67021-39829. H. Reisizadeh and M. Hong are partly supported by NSF grants ECCS-2426064, and a grant from Open Philanthropy TechnicalAI Safety. The work of J. Jia and S. Liu was supported in part by the Amazon Research Award for AI in Information Security, the NSF Award IIS-2504263, the Open Philanthropy Research Award, the Schmidt Sciences Trustworthy AI Program, and the Center for AI Safety (CAIS) Compute Award. K. Chang is partly supported by NIH 1OT2OD038003-01, NSF 2331966 and Open Philanthropy.

7 Limitations

This work introduces BLUR, a bi-level optimization approach for unlearning in LLMs, which is essential for ethical AI development, data privacy compliance (e.g., GDPR), and mitigating harmful outputs. By prioritizing unlearning over retention, BLUR effectively removes sensitive or biased data while maintaining model utility, enhancing AI safety and fairness. However, verifying complete and irreversible unlearning remains a challenge, requiring future research on rigorous evaluation methods. As AI becomes more integrated into sensitive domains like healthcare, finance, and law, unlearning techniques like BLUR can help prevent misinformation, protect privacy, and improve public trust in AI systems.

References

Clark Barrett, Brad Boyd, Elie Bursztein, Nicholas Carlini, Brad Chen, Jihye Choi, Amrita Roy Chowdhury, Mihai Christodorescu, Anupam Datta, Soheil Feizi, and 1 others. 2023. Identifying and mitigating the security risks of generative ai. *Foundations and Trends® in Privacy and Security*, 6(1):1–52.

Reuben Binns. 2018. Fairness in machine learning: Lessons from political philosophy. In *Conference on fairness, accountability and transparency*, pages 149–159. PMLR.

Zhiqi Bu, Xiaomeng Jin, Bhanukiran Vinzamuri, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, and Mingyi Hong. 2024. Unlearning as multi-task optimization: A normalized gradient difference approach with an adaptive learning rate. *arXiv preprint arXiv:2410.22086*.

Tianshi Che, Yang Zhou, Zijie Zhang, Lingjuan Lyu, Ji Liu, Da Yan, Dejing Dou, and Jun Huan. 2023. Fast federated machine unlearning with nonlinear functional theory. In *International conference on machine learning*, pages 4241–4268. PMLR.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2022. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 499–513.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. 2022. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. *Advances in Neural Information Processing Systems*, 35:26698–26710.

Stephan Dempe, Nguyen Dinh, Joydeep Dutta, and Tanushree Pandit. 2021. Simple bilevel programming and extensions. *Mathematical Programming*, 188:227–253.

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

Joydeep Dutta and Tanushree Pandit. 2020. Algorithms for simple bilevel programming. *Bilevel Optimization: Advances and Next Challenges*, pages 253–291.

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

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

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. 2023. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. 2018. Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning*, pages 1568–1577. PMLR.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. 2023. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2426–2436.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, and 1 others. 2021. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*, 10:8–9.

GDPR GDPR. 2016. General data protection regulation. *Regulation (EU)*, 679.

Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. 2024. Jogging the memory of unlearned models through targeted relearning attacks. In *ICML 2024 Workshop on Foundation Models in the Wild*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

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

Kaiyi Ji, Junjie Yang, and Yingbin Liang. 2021. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pages 4882–4892. PMLR.

Jinghan Jia, Jiancheng Liu, Yihua Zhang, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. 2024. Wagle: Strategic weight attribution for effective and modular unlearning in large language models. *arXiv preprint arXiv:2410.17509*.

Ruichen Jiang, Nazanin Abolfazli, Aryan Mokhtari, and Erfan Yazdandoost Hamedani. 2023. A conditional gradient-based method for simple bilevel optimization with convex lower-level problem. In *International Conference on Artificial Intelligence and Statistics*, pages 10305–10323. PMLR.

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

Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Collective Intelligence Conference*, pages 12–24.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, and 1 others. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*.

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

Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. 2020. Federated unlearning. *arXiv preprint arXiv:2012.13891*.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. 2024. Towards safer large language models through machine unlearning. *arXiv preprint arXiv:2402.10058*.

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

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

Dougal Maclaurin, David Duvenaud, and Ryan Adams. 2015. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024. Tofu: A task of fictitious unlearning for llms. *Preprint*, arXiv:2401.06121.

Fabio Motoki, Valdemar Pinho Neto, and Victor Rodrigues. 2023. More human than human: Measuring chatgpt political bias. *Available at SSRN 4372349*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Jeffrey Rosen. 2011. The right to be forgotten. *Stan. L. Rev. Online*, 64:88.

Shoham Sabach and Shimrit Shtern. 2017. A first order method for solving convex bilevel optimization problems. *SIAM Journal on Optimization*, 27(2):640–660.

Bhavika Sachdeva, Harshita Rathee, Arun Sharma, Witold Wydmański, and 1 others. 2024. Machine unlearning for recommendation systems: An insight. *arXiv preprint arXiv:2401.10942*.

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

Lingqing Shen, Nam Ho-Nguyen, and Fatma Kılınç-Karzan. 2023. An online convex optimization-based framework for convex bilevel optimization. *Mathematical Programming*, 198(2):1519–1582.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. 2024. Muse: Machine unlearning six-way evaluation for language models. *arXiv preprint arXiv:2407.06460*.

Ilia Shumailov, Jamie Hayes, Eleni Triantafillou, Guillermo Ortiz-Jimenez, Nicolas Papernot, Matthew Jagielski, Itay Yona, Heidi Howard, and Eugene Bagdasaryan. 2024. Ununlearning: Unlearning is not sufficient for content regulation in advanced generative ai. *arXiv preprint arXiv:2407.00106*.

Bradly Stadie, Lunjun Zhang, and Jimmy Ba. 2020. Learning intrinsic rewards as a bi-level optimization problem. In *Conference on Uncertainty in Artificial Intelligence*, pages 111–120. PMLR.

Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, and 1 others. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. 2022. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319. IEEE.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jiulin Wang, Xu Shi, and Rujun Jiang. 2024. Near-optimal convex simple bilevel optimization with a bisection method. In *International Conference on Artificial Intelligence and Statistics*, pages 2008–2016. PMLR.

Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. 2023. Unveiling the implicit toxicity in large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Kun Wu, Jie Shen, Yue Ning, Ting Wang, and Wendy Hui Wang. 2023. Certified edge unlearning for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2606–2617.

Mimee Xu, Jiankai Sun, Xin Yang, Kevin Yao, and Chong Wang. 2023. Netflix and forget: Efficient and exact machine unlearning from bi-linear recommendations. *arXiv preprint arXiv:2302.06676*.

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

Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. 2023. When demonstrations meet generative world models: A maximum likelihood framework for offline inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 36:65531–65565.

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

Yimeng Zhang, Xin Chen, Jinghan Jia, Yihua Zhang, Chongyu Fan, Jiancheng Liu, Mingyi Hong, Ke Ding, and Sijia Liu. 2024b. Defensive unlearning with adversarial training for robust concept erasure in diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

A The Preliminaries

Lemma 1. *For any $\boldsymbol{\theta} \in \mathbb{R}^d$, we can write*

(a) $\langle \nabla f(\boldsymbol{\theta}), u(\boldsymbol{\theta}) \rangle = \gamma \|\nabla f(\boldsymbol{\theta})\|^2$.

Also, under Assumption 1, for every $\boldsymbol{\theta} \in \mathbb{R}^d$, we get

(b) $\|u(\boldsymbol{\theta})\| \leq C_1$,

(c) $\xi(\boldsymbol{\theta}) \|\nabla f(\boldsymbol{\theta})\|^2 \leq \gamma \|\nabla f(\boldsymbol{\theta})\|^2 + C \|\nabla f(\boldsymbol{\theta})\|$,

(d) $\xi(\boldsymbol{\theta}) \langle \nabla f(\boldsymbol{\theta}), u(\boldsymbol{\theta}) \rangle \leq \gamma^2 \|\nabla f(\boldsymbol{\theta})\|^2 + C\gamma \|\nabla f(\boldsymbol{\theta})\|$,

(e) $\langle \nabla r(\boldsymbol{\theta}), u(\boldsymbol{\theta}) \rangle \geq \|u(\boldsymbol{\theta})\|^2 - \gamma^2 \|\nabla f(\boldsymbol{\theta})\|^2 - C\gamma \|\nabla f(\boldsymbol{\theta})\|$,

where $\xi(\boldsymbol{\theta}) := \frac{\gamma \|\nabla f(\boldsymbol{\theta})\|^2 - \langle \nabla f(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta}) \rangle}{\|\nabla f(\boldsymbol{\theta})\|^2}$.

Proof. From the definition of $\xi(\boldsymbol{\theta})$ and $u(\boldsymbol{\theta})$ in (10), we have

$$\begin{aligned}
&\langle \nabla f(\boldsymbol{\theta}), u(\boldsymbol{\theta}) \rangle \\
&= \langle \nabla f(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta}) \rangle + \xi(\boldsymbol{\theta}) \|\nabla f(\boldsymbol{\theta})\|^2 \\
&= \langle \nabla f(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta}) \rangle + \gamma \|\nabla f(\boldsymbol{\theta})\|^2 - \langle \nabla r(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta}) \rangle \\
&= \gamma \|\nabla f(\boldsymbol{\theta})\|^2,
\end{aligned}$$

that completes the proof of part (a). From the definition of $u(\boldsymbol{\theta})$ and the triangle inequality, we can write

$$\begin{aligned}
&\|\xi(\boldsymbol{\theta}) \nabla f(\boldsymbol{\theta}) + \nabla r(\boldsymbol{\theta})\| \\
&\leq |\xi(\boldsymbol{\theta})| \|\nabla f(\boldsymbol{\theta})\| + \|\nabla r(\boldsymbol{\theta})\| \\
&\leq \gamma \|\nabla f(\boldsymbol{\theta})\| + \frac{|\langle \nabla f(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta}) \rangle|}{\|\nabla f(\boldsymbol{\theta})\|} + \|\nabla r(\boldsymbol{\theta})\| \\
&\overset{(a)}{\leq} \gamma \|\nabla f(\boldsymbol{\theta})\| + 2 \|\nabla r(\boldsymbol{\theta})\| \leq (2 + \gamma) C,
\end{aligned}$$

where (a) follows from the Cauchy-Schwartz inequality and the last step holds due to the Assumption 1.

To prove part (c), using the Cauchy-Schwartz inequality, we get

$$\begin{aligned}
&\xi(\boldsymbol{\theta}) \|\nabla f(\boldsymbol{\theta})\|^2 \\
&= \gamma \|\nabla f(\boldsymbol{\theta})\|^2 - \langle \nabla f(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta}) \rangle \\
&\leq \gamma \|\nabla f(\boldsymbol{\theta})\|^2 + \|\nabla f(\boldsymbol{\theta})\| \cdot \|\nabla r(\boldsymbol{\theta})\| \\
&\leq \gamma \|\nabla f(\boldsymbol{\theta})\|^2 + C \|\nabla f(\boldsymbol{\theta})\|, \qquad (19)
\end{aligned}$$

where the last inequality holds from Assumption 1. Now, we prove part (d). From (10), we have

$$\begin{aligned}
&\xi(\boldsymbol{\theta}) \langle \nabla f(\boldsymbol{\theta}), u(\boldsymbol{\theta}) \rangle \\
&= \xi(\boldsymbol{\theta}) \langle \nabla f(\boldsymbol{\theta}), \xi(\boldsymbol{\theta}) \nabla f(\boldsymbol{\theta}) + \nabla r(\boldsymbol{\theta}) \rangle \\
&= \xi(\boldsymbol{\theta}) \left(\xi(\boldsymbol{\theta}) \|\nabla f(\boldsymbol{\theta})\|^2 + \langle \nabla f(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta}) \rangle \right) \\
&= \xi(\boldsymbol{\theta}) \gamma \|\nabla f(\boldsymbol{\theta})\|^2 \\
&\quad + \xi(\boldsymbol{\theta}) \left(- \langle \nabla f(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta}) \rangle + \langle \nabla f(\boldsymbol{\theta}), \nabla r(\boldsymbol{\theta}) \rangle \right) \\
&= \xi(\boldsymbol{\theta}) \gamma \|\nabla f(\boldsymbol{\theta})\|^2. \qquad (20)
\end{aligned}$$

This together with part (c) leads us to part (d). Finally, to show part (e), from part (d), we get

$$\begin{aligned}
&\langle \nabla r(\boldsymbol{\theta}), u(\boldsymbol{\theta})\rangle \\
&= \langle u(\boldsymbol{\theta}) - \xi(\boldsymbol{\theta})\nabla f(\boldsymbol{\theta}), u(\boldsymbol{\theta})\rangle \\
&= \|u(\boldsymbol{\theta})\|^2 - \xi(\boldsymbol{\theta})\langle \nabla f(\boldsymbol{\theta}), u(\boldsymbol{\theta})\rangle \\
&\geq \|u(\boldsymbol{\theta})\|^2 - \gamma^2\|\nabla f(\boldsymbol{\theta})\|^2 - C\gamma\|\nabla f(\boldsymbol{\theta})\|.
\end{aligned}$$

□

Lemma 2. *For any vectors* $\{\mathbf{u}_i\}_{i=1}^n \in \mathbb{R}^d$*, we get* $\|\sum_{i=1}^n \mathbf{u}_i\|^2 \leq n\sum_{i=1}^n \|\mathbf{u}_i\|^2$.

B Proof of Theorem 1

Using the fact that f is L_f-Lipschitz from Assumption 1, we get

$$\begin{aligned}
&f(\boldsymbol{\theta}(t+1)) - f(\boldsymbol{\theta}(t)) \\
&\leq \langle \nabla f(\boldsymbol{\theta}(t)), \boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\rangle + \frac{L_f}{2}\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\|^2 \\
&= -\eta\langle \nabla f(\boldsymbol{\theta}(t)), u(\boldsymbol{\theta}(t))\rangle + \frac{L_f}{2}\eta^2\|u(\boldsymbol{\theta}(t))\|^2 \\
&\overset{(a)}{=} -\eta\gamma\|\nabla f(\boldsymbol{\theta}(t))\|^2 + \frac{L_f}{2}\eta^2\|u(\boldsymbol{\theta}(t))\|^2 \\
&\leq -\eta\gamma\|\nabla f(\boldsymbol{\theta}(t))\|^2 + \frac{L_f}{2}\eta^2 C_1^2, \qquad (21)
\end{aligned}$$

where (a) follows from Lemma 1-(a), and the last step holds due to Lemma 1-(b). Applying a telescopic summation in (21), we can write

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}(t))\|^2 \\
&\leq \frac{1}{T\eta\gamma}\left(f(\boldsymbol{\theta}(0)) - f(\boldsymbol{\theta}(T))\right) + \frac{L_f}{2\gamma}\eta C_1^2 \\
&\leq \frac{2C}{T\eta\gamma} + \frac{L_f}{2\gamma}\eta C_1^2,
\end{aligned}$$

where the last step follows from Assumption 1. That completes the proof of the first claim of the theorem. Similarly, since r is L_r-Lipschitz from Assumption 1, we arrive at

$$\begin{aligned}
&r(\boldsymbol{\theta}(t+1)) - r(\boldsymbol{\theta}(t)) \\
&\leq \langle \nabla r(\boldsymbol{\theta}(t)), \boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\rangle + \frac{L_r}{2}\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\|^2 \\
&= -\eta\|u(\boldsymbol{\theta}(t))\|^2 + \eta\xi(\boldsymbol{\theta}(t))\langle \nabla f(\boldsymbol{\theta}(t)), u(\boldsymbol{\theta}(t))\rangle \\
&\quad + \frac{L_r}{2}\eta^2\|u(\boldsymbol{\theta}(t))\|^2 \qquad (22) \\
&\leq -\frac{\eta}{2}\|u(\boldsymbol{\theta}(t))\|^2 \\
&\quad + \eta\left(\gamma^2\|\nabla f(\boldsymbol{\theta}(t))\|^2 + C\gamma\|\nabla f(\boldsymbol{\theta}(t))\|\right). \qquad (23)
\end{aligned}$$

Applying a telescopic summation in (22), we get

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\|u(\boldsymbol{\theta}(t))\|^2 \\
&\leq \frac{2}{\eta T}\left(r(\boldsymbol{\theta}(0)) - r(\boldsymbol{\theta}(t))\right) \\
&\quad + \frac{2}{T}\left(\gamma^2\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}(t))\|^2 + \gamma\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}(t))\|\right) \\
&\leq \frac{4C}{\eta T} \\
&\quad + \frac{2}{T}\left(\gamma^2\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}(t))\|^2 + \gamma\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}(t))\|\right). \qquad (24)
\end{aligned}$$

For the second term in (24), using Lemma 2, we can write

$$\begin{aligned}
&\gamma^2\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}(t))\|^2 + \gamma\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}(t))\| \\
&\leq \gamma^2\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}(t))\|^2 + \gamma\sqrt{T\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}(t))\|^2} \\
&\leq \frac{2C\gamma}{\eta} + \frac{L_f\gamma}{2}\eta C_1^2 T + \gamma\sqrt{\frac{2CT}{\eta\gamma} + \frac{L_f}{2\gamma}\eta C_1^2 T^2}, \qquad (25)
\end{aligned}$$

where the last step holds due to (12). Finally, for $T \geq \frac{4C}{L_f C_1^2\eta^2}$, from (24)-(25), we conclude

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\|u(\boldsymbol{\theta}(t))\|^2 \\
&\leq \frac{4C}{\eta T}(1+\gamma) + L_f\gamma\eta C_1^2 + 2\gamma C_1\sqrt{\frac{L_f}{\gamma}\eta}.
\end{aligned}$$

C Additional Experiment Results and Details

In the following, we provide the corresponding forget loss functions discussed in Section 2:

$$\ell_{\text{GA}}(y \mid x; \boldsymbol{\theta}) = \log \pi(y \mid x; \boldsymbol{\theta}), \qquad (26)$$

$$\ell_{\text{NPO},\beta}(y \mid x; \boldsymbol{\theta}) = \frac{2}{\beta}\log\left(1 + \left(\frac{\pi(y \mid x; \boldsymbol{\theta})}{\pi(y \mid x; \boldsymbol{\theta}_0)}\right)^{\beta}\right), \qquad (27)$$

$$\ell_{\text{SimNPO},\beta,\alpha}(y \mid x; \boldsymbol{\theta}) = -\frac{2}{\beta}\log\sigma\left(-\frac{\beta}{|y|}\log\pi(y \mid x; \boldsymbol{\theta}) - \alpha\right), \qquad (28)$$

$$\ell_{\text{RMU},f}(y \mid x; \boldsymbol{\theta}) = \|M_i(x; \boldsymbol{\theta}) - c \cdot \mathbf{u}\|_2^2, \qquad (29)$$

Unlearning Method	Retain Loss	Forget Loss
GA (Maini et al., 2024)	N/A	(26)
GradDiff (Liu et al., 2022)	(2)	(26)
NPO (Zhang et al., 2024a)	(2)	(27)
SimNPO (Fan et al., 2024)	(2)	(28)

Table 6: Summary of unlearning methods with their retain and forget losses.

Method	η	β	γ	α	λ
GA	10^{-5}	-	-	-	-
GradDiff	10^{-5}	-	-	-	1.0
NPO	10^{-5}	0.1	-	-	1.0
SimNPO	10^{-5}	2.5	-	2.0	0.15
BLUR–NPO	10^{-5}	0.1	1.0	-	-

Table 7: Hyperparamters for various unlearning methods on TOFU benchmark

Here, $\pi(y \mid x; \boldsymbol{\theta}_0)$ denotes the reference distribution of the pre-trained model, $|y|$ denotes the response length, $\beta \geq 0$ is a sharpness parameter, $\alpha \geq 0$ is a margin parameter in SimNPO, $\mathbf{u}$ is a fixed random unit vector, and c controls the scaling of representation perturbations. Using the update rule in (4), and applying different choices for the forget and retain losses, we derive various unlearning methods, as summarized in **Table 6**.

C.1 Experiment Setups

Computational Configurations & Complexity. All experiments are conducted on 8 NVIDIA A100 GPUs. BLUR is designed as a **first-order** method, similar in memory complexity to all other baselines. It does not require storing or computing higher-order derivatives, Jacobians, or Hessians. Our bilevel formulation in (6) is resolved through a lightweight gradient projection, where we compute the component of the retain gradient orthogonal to the forget gradient. This projection is performed using standard inner product and norm computations, and introduces no additional memory overhead beyond that of standard backpropagation.

Further, each update step in BLUR introduces only two additional scalar operations: (1) an inner product between the forget and retain gradients, and (2) a scalar division by the squared norm of the forget gradient. These operations are computationally negligible relative to the forward and backward passes through the model. Moreover, no auxiliary optimization loops or nested gradient steps are required. Next, we present the hyperparameters used in each unlearning task.

TOFU. We conduct experiments for 10 epochs using a learning rate of 10^{-5} and a batch size of 32. A grid search is performed over the range $[0.5, 2]$ for γ, and the parameter β is searched within $[0.05, 0.2]$, with the final value of 0.1. We set $\lambda = 1$ for GradDiff and NPO, and $\lambda = 2.5$ for SimNPO. The hyperparameters β for NPO and SimNPO are set to 0.1 and 2.5, respectively. For the SimNPO unlearning scheme, the hyperparameter α is fixed at 0.125. We summarized the hyperparameters in **Table 7**.

MUSE. We train our algorithm BLUR–NPO for 10 epochs with a constant learning rate of 2.5×10^{-5} for the news dataset and 10^{-5} for the books dataset, the batch size of 32, and an input length of $2,048$ tokens. We perform a grid search for γ in the range of $[0.8, 1.2]$ and set $\gamma = 1.0$ as the final value. For the NPO loss in (27), β is set to 0.05 for the news dataset and 0.4 for the books dataset. The evaluation pipelines strictly follow the setup detailed by (Shi et al., 2024). Further, we evaluate the model's performance after the unlearning process and select the optimal model as the final result. We use a constant learning rate of 2.5×10^{-5} for news and 5×10^{-6} for books datasets in NGDiff, with $\beta = 0.1$ for both corpora. The hyperparameters for other unlearning methods GA, GradDiff, NPO, and SimNPO are set according to the works of (Shi et al., 2024; Fan et al., 2024). **Table 8** summarizes our method's optimal combination of the hyperparameters, determined through grid search. We used the hyperparameters reported in the corresponding papers for other unlearning methods.

Method	Dataset	η	β	γ	α	λ
GA	News/Books	10^{-5}	-	-	-	0
GradDiff	News/Books	10^{-5}	-	-	-	1
NGDiff	News	2.5×10^{-5}	0.1	-	-	-
	Books	5×10^{-6}	0.1	-	-	-
NPO	News/Books	10^{-5}	0.1	-	-	1
SimNPO	News	10^{-5}	0.7	-	0.0	0.1
	Books	10^{-5}	0.75	-	0.0	0.1
BLUR–NPO	News	2.5×10^{-5}	0.05	1.0	-	-
	Books	10^{-5}	0.4	1.0	-	-

Table 8: Hyperparameters for various unlearning methods on MUSE benchmark

WMDP. We run the experiments for BLUR–

Method	η	β	γ	α	λ	c
RMU	5×10^{-5}	-	-	-	1200	6.5
NPO	10^{-5}	0.1	-	-	1.0	-
SimNPO	10^{-5}	0.1	-	0.0	1.0	-
NGDiff–RMU	10^{-5}	-	-	-	-	6.5
BLUR–NPO	2×10^{-6}	0.005	1.0	-	-	-
BLUR–RMU	4×10^{-2}	-	0.00125	-	-	6.5

Table 9: Hyperparamters for various unlearning methods on WMDP benchmark

NPO using a constant learning rate of 2×10^{-6} and a batch size of 4. A grid search is performed for γ within the range [0.5,1.5], and β in (27) is explored within the range $[0.001, 0.01]$. As for the RMU and BLUR–RMU algorithms, we follow the implementation details provided in WMDP (Li et al., 2024). More precisely, for BLUR–RMU, we conduct a grid search for γ over the range $[0.001, 0.002]$ and use 0.00125 as the final γ. We also employ a constant learning rate of 4×10^{-2} and train for 150 steps for BLUR–RMU. We exploit `lm-evaluation-harness v0.4.2` (Gao et al., 2021) to standardize prompts. We set the experimental parameters for RMU, NPO, and SimNPO, as described in (Li et al., 2024; Jia et al., 2024; Fan et al., 2024). **Table 9** summarizes all the hyperparameters used for the unlearning.

We note that all other baselines applied *extensive* hyperparameter fine-tuning and reported their **best-performing parameter** settings. We adopt their reported optimal settings directly, ensuring that each baseline is compared at its best-reported configuration without additional tuning on our end.

C.2 Additional Experiment Results

Here, we discuss our experimental results further. As **Fig** C.1 shows, the update direction $\hat{u}(\boldsymbol{\theta})$ cannot consistently prioritize the forget loss over the retain loss, even for various values of λ. This further corroborates that the regularized problem in (1) fails to achieve a proper balance between the forget and retain losses. **Fig** C.2 show that the retain and forget gradients conflict over the unlearning steps across various unlearning schemes and values of λ.

Ablation Studies of BLUR–NPO on MUSE-News Corpus.

The proposed direction $u(\boldsymbol{\theta})$ in (10) exploits a hyperparameter γ adjusting the amplitude of the forget gradient. Moreover, for BLUR–NPO, we use NPO loss for the forget objective, where the hyperparameter β regulates the intensity of unlearning. As $\beta \to 0$, NPO loss converges to the GA loss. **Fig** C.3 illustrates the performance of KnowMem on the forget and retain datasets of BLUR–NPO for various values of β and γ on MUSE-News using the LLaMA2-7B model with two learning rates $\eta = 10^{-5}$ and $\eta = 2.5\times10^{-5}$. As observed, a large value of β fails to unlearn the forget set while maintaining good performance on the retain set. Moreover, the learning rate $\eta = 10^{-5}$ is not large enough to unlearn the forget set while preserving the model utility, whereas a larger learning rate $\eta = 2.5\times10^{-5}$ with a relatively small $\beta = 0.05$ and $\gamma = 1.0$ achieves both objectives simultaneously. The unlearning metrics for this optimal setup are highlighted with red rectangles in **Fig.** C.3.

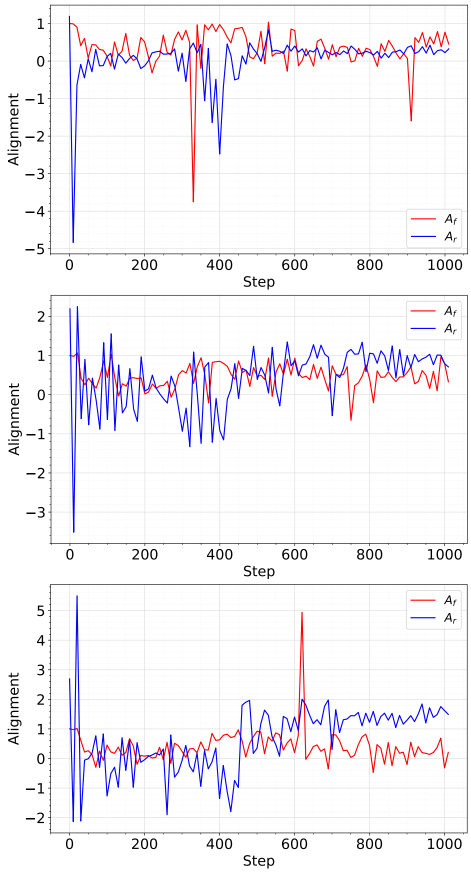
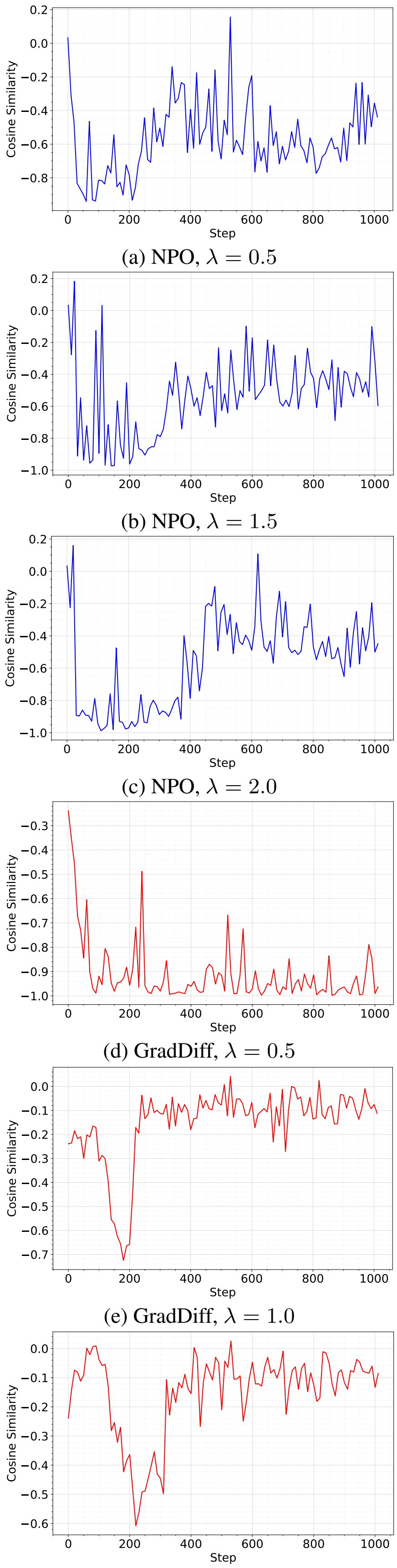


Figure C.1: Alignment values of forget and retain objectives as defined in (5) on MUSE-News dataset using LLaMa2-7B model vs. step with $\lambda = 0.5$ (top), $\lambda = 1.5$ (middle), and $\lambda = 2$ (bottom).



(a) NPO, $\lambda = 0.5$

(b) NPO, $\lambda = 1.5$

(c) NPO, $\lambda = 2.0$

(d) GradDiff, $\lambda = 0.5$

(e) GradDiff, $\lambda = 1.0$

(f) GradDiff, $\lambda = 1.5$

Figure C.2: Cosine similarity of the gradient forget and retain losses using NPO and GradDiff methods on MUSE-News dataset using LLaMa2-7B model vs. step.

β \ γ	0.8	0.9	1.0	1.1	1.2
0.2	47.4	44.8	47.0	42.9	47.3
0.15	45.7	46.4	43.0	48.0	40.3
0.1	41.8	40.6	37.9	39.3	39.2
0.05	28.2	35.1	36.5	32.3	35.7
$\to 0$	0.0	0.0	0.0	0.0	0.0

(a) KnowMem on *forget* dataset with $\eta = 10^{-5}$.

β \ γ	0.8	0.9	1.0	1.1	1.2
0.2	42.0	43.2	44.7	43.8	42.4
0.15	46.0	43.3	41.8	44.7	39.2
0.1	42.1	42.1	40.5	43.5	44.4
0.05	35.9	34.6	37.7	33.4	26.4
$\to 0$	0.0	0.0	0.0	0.0	0.0

(b) KnowMem on *retain* dataset with $\eta = 10^{-5}$.

β \ γ	0.8	0.9	1.0	1.1	1.2
0.2	50.5	40.7	38.0	41.8	41.3
0.15	40.4	39.6	42.3	42.2	30.8
0.1	47.3	39.0	39.9	31.0	35.0
0.05	32.9	34.2	29.0	46.7	22.1
$\to 0$	0.0	0.0	0.0	0.0	0.0

(c) KnowMem on *forget* dataset with $\eta = 2.5 \times 10^{-5}$.

β \ γ	0.8	0.9	1.0	1.1	1.2
0.2	38.2	40.7	43.4	44.8	40.6
0.15	40.2	41.9	43.7	38.0	42.6
0.1	42.0	46.0	39.5	44.7	40.4
0.05	27.1	30.3	46.7	43.6	40.1
$\to 0$	0.0	0.0	0.0	0.0	0.0

(d) KnowMem on *retain* dataset with $\eta = 2.5 \times 10^{-5}$.

Figure C.3: KnowMem values on *forget* and *retain* datasets using BLUR–NPO unlearning method, LLaMA2-7B model, and MUSE-News corpus under two learning rates $\eta = 10^{-5}$ and $\eta = 2.5 \times 10^{-5}$ with different combinations of hyperparameters of β and γ.