Detoxification via Gradient Surgery

Qirui Zheng
Halıcıoğlu Data Science Institute
University of California, San Diego
San Diego, CA 92092
q7zheng@ucsd.edu

Jun-Kun Wang
Halıcıoğlu Data Science Institute
University of California, San Diego
San Diego, CA 92092
jkw005@ucsd.edu

Abstract

Large language models can retain harmful behaviors from toxic training data, making detoxification an important challenge for machine unlearning. A central difficulty is that unlearning must simultaneously suppress toxic behavior and preserve general language modeling ability, creating inherently conflicting optimization objectives. We cast this problem as multi-task learning, where a forget objective and a retain objective interact through conflicting gradients. We instantiate this idea for GPT-2-scale causal language models with two unlearning families GradDiff and idkDPO. Experiments on detoxification show that gradient surgery (PCGrad) consistently improves the toxicity–utility trade-off relative to the corresponding baselines. In particular, PCGrad reduces toxicity in both objective families, while yielding the strongest overall results when paired with idkDPO, where it also preserves low WikiText perplexity. Membership inference and weight-space analyses further suggest that PCGrad improves forgetting behavior by steering optimization away from destructive interference rather than by radically altering the underlying objective. These findings suggest that explicit conflict-aware optimization is a practical and general strategy for more effective and utility-preserving language model unlearning. Code is available at: https://github.com/Qz07/ToxiGS.

1 Introduction

Large language models (LLMs) acquire broad capabilities by training on large scale corpora, but these corpora inevitably contain harmful, toxic, or otherwise undesirable behaviors. As a result, after standard fine-tuning, pretrained models can still produce toxic generations under adversarial prompts. Retraining foundation models from scratch on carefully filtered data is often cost and computationally infeasible, motivating growing interest in machine unlearning which is a post training methods that selectively suppress undesirable training data without discarding the useful knowledge acquired during pretraining. In practice, however, unlearning is fundamentally difficult because the model must simultaneously satisfy two competing goals. It should forget harmful data on a targeted subset of data while retaining general language modeling ability on the remainder.

This is relevant in language model detoxification. Aggressively optimizing a forget objective can reduce toxicity, but often at the cost of severe degradation in perplexity and downstream model quality. Such failures are commonly manifested as **catastrophic collapse**, where updates intended to erase harmful behavior interfere with parameters that also support general-purpose language understanding and generation. At the same time, successful unlearning should not only reduce harmful outputs, but also weaken residual evidence that the model still memorizes forget-set examples, as measured by privacy-oriented tests such as membership inference. Thus, practical machine unlearning must balance at least three desired effectiveness in suppressing harmful behavior, utility preservation on non-forget data, and privacy improvement with respect to the forgotten subset.

In this work, we argue that this trade-off is best viewed through the lens of multi-task learning. Rather than treating unlearning as a single-objective optimization problem, we model it as the interaction

between two competing objectives: a forgetting objective that suppresses toxic or target behavior, and a retention objective that preserves useful language modeling competence. This perspective suggests a natural role for methods that explicitly mitigate gradient conflict. We adopt **Gradient Surgery** (PCGrad), originally proposed for multi-task learning, as a simple and general mechanism for resolving interference between forget and retain updates[Yu et al., 2020a]. Intuitively, when the unlearning gradient and the retention gradient point in conflicting directions, PCGrad projects away the destructive component, allowing the model to make progress on forgetting while reducing damage to retained knowledge.

We instantiate this idea in the context of detoxification for GPT-2-scale causal language models. Our experiments study two representative unlearning families: a gradient-difference style objective and a preference-based "I don't know" objective (idkDPO) [Maini et al., 2024]. In both cases, we augment training with PCGrad to explicitly handle gradient conflict between forgetting toxic behavior and preserving next-token prediction performance on retain data. We evaluate these methods along three axes: (1) toxicity reduction, (2) utility preservation via WikiText word perplexity, and (3) privacy/unlearning quality via membership inference statistics. We additionally examine pairwise distances in weight space to understand how gradient surgery changes the geometry of the learned solution.

2 Related Work

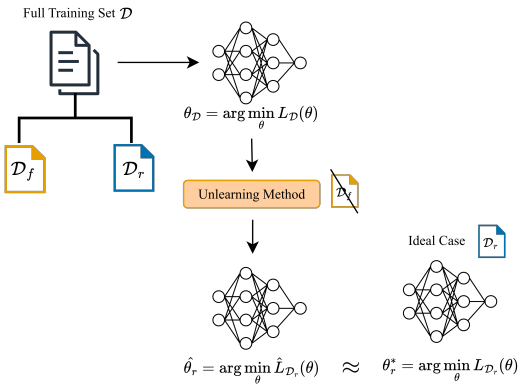

Figure 1: Pipeline overview of machine unlearning procedures.

Machine unlearning: Machine unlearning aims to remove the influence of designated training data from a model without requiring full retraining. In the context of generative models and large language models, this problem is especially challenging because undesirable training data can be available at inference time through memorization, harmful continuations, or privacy leakage. Recent work has therefore emphasized post hoc unlearning methods that operate directly on pretrained generative models while attempting to preserve their general capabilities [Maini et al., 2024, Zhang et al., 2024]. A central concern in these works is that effective forgetting often conflicts with retained model utility: aggressively updating the model to suppress target behaviors can damage language modeling performance, factual knowledge, and output quality. Figure 1, depicts the pipeline for machine unlearning, as the unlearned model tries to estimate the parameters in the ideal case where the model is just trained on the retain set.

Gradient-based unlearning and catastrophic collapse: A common practical strategy for LLM unlearning is to optimize a forget objective directly, often via gradient ascent on the loss over forget examples, sometimes together with a retain-side objective that anchors the model on non-forget data. Although simple and effective in some settings, such approaches are prone to instability and catastrophic collapse, where the utility of the model degrades dramatically as the pressure of forgetting increases [Zhang et al., 2024]. This failure mode is particularly relevant in detoxification settings, where the goal is not merely to worsen likelihood on forget examples, but to selectively suppress toxic generations while maintaining fluent and useful behavior elsewhere. Preference-based

formulations such as Negative Preference Optimization (NPO) were introduced to improve this trade-off by slowing the progression toward collapse relative to naive gradient ascent [Zhang et al., 2024].

Gradient surgery as a view of unlearning: Several recent works suggest that the geometry of gradients is central to successful unlearning. Bae et al. [2023] show that in generative models, forgetting can be framed through the interaction between removal and retention gradients, and propose a gradient-surgery procedure that projects updates to reduce harmful interference. This perspective is especially appealing for language-model unlearning, where forgetting and retention are not independent goals but inherently competing objectives defined over the same parameters. More recently, Zhou et al. [2025] formalize utility-preserving machine unlearning as a constrained optimization problem and show that the resulting solution can be interpreted as unilateral gradient surgery. Their analysis reinforces the view that effective unlearning should not be treated as a single-objective optimization problem, but rather as one that must explicitly balance deletion and preservation.

Multi-task optimization and PCGrad: PCGrad [Yu et al., 2020b] was proposed as a simple and effective method for mitigating gradient conflict by projecting one task gradient onto the normal plane of another whenever their inner product is negative. Although originally developed for standard multi-task learning, its underlying motivation directly applies to machine unlearning. In our setting, the forget objective encourages the model to suppress harmful or memorized behaviors, while the retain objective preserves useful next-token prediction behavior on non-forget data. These objectives naturally induce gradient conflict, particularly when forget examples overlap semantically or distributionally with benign language. By treating unlearning as multi-task optimization and applying PCGrad to the forget and retain losses, our method provides a principled mechanism for reducing destructive interference. Conceptually, this differs from coefficient tuning or heuristic regularization: the update is modified only when the two objectives are in direct conflict, allowing the model to forget more selectively while preserving utility.

Detoxification and harmful generation: Our work is further connected to the literature on toxicity mitigation and harmful text generation. ToxiGen [Hartvigsen et al., 2022] highlights the difficulty of detecting and mitigating subtle, adversarial, and implicit hate speech, and has become a standard benchmark for safety-oriented evaluation. Unlike broad detoxification strategies based on filtering or generic alignment, machine unlearning offers a more targeted intervention: the model can be trained to remove the influence of harmful data or suppress harmful continuations while retaining general knowledge and fluency. This framing is particularly important because overly aggressive safety interventions can erase benign capabilities or distort model behavior on non-toxic content.

Evaluation of utility and privacy in unlearning: A growing body of work argues that unlearning should be evaluated holistically rather than through forget-set performance alone. TOFU emphasizes that successful unlearning should approximate the behavior of a model that was never trained on the deleted data, while also preserving utility on retained tasks [Maini et al., 2024]. More generally, broad capability evaluations such as MMLU are often used to test whether post hoc interventions preserve downstream competence [Hendrycks et al., 2021]. At the same time, privacy-oriented metrics remain essential because memorization can persist even when behavioral forgetting appears successful. Prior work on membership inference shows that differences in model likelihood between training and non-training examples can reveal training-set membership, linking overfitting and memorization to privacy risk [Yeom et al., 2018].

3 Method

3.1 Problem Setting

We study detoxification as a targeted machine unlearning problem for causal language models. Let π_θ denote a language model with parameters θ. Each training example consists of a prompt completion pair (x, y), where x is the prompt and $y = (y_{p+1}, \ldots, y_T)$ is the generated continuation. Following the standard causal language modeling setup, the prompt tokens serve only as conditioning context,

while supervision is applied exclusively to the completion tokens:

$$L_{\mathrm{NTP}}(\theta; x, y) = -\sum_{t=p+1}^{T} \log p_\theta(y_t \mid x, y_{p+1:t-1}).$$

This masking is important in our setting because the objective is not to erase knowledge of toxic prompts themselves, but rather to suppress harmful continuations while preserving general language modeling ability.

We partition the training set into a forget subset $\mathcal{D}_f$ and a retain subset $\mathcal{D}_r$. The forget subset contains toxic prompt completion pairs that the model should unlearn, while the retain subset contains non-toxic or utility-preserving examples used to regularize the model against catastrophic degradation. In our experiments, the base model is a GPT-2 language model fine-tuned on a corpus of approximately 250k examples, yielding the starting checkpoint from which all unlearning methods are derived from.

Our goal is to learn parameters θ such that: (i) generations conditioned on toxic prompts are suppressed or redirected to safe style behavior, (ii) performance on benign language modeling is preserved, and (iii) privacy leakage on forgotten examples is reduced. We formalize this as a two-task optimization problem, where unlearning and retention induce distinct gradient signals that may conflict.

3.2 Training Data

Our detoxification data is from toxic language examples in the ToxiGen benchmark [Hartvigsen et al., 2022], which was designed to capture both adversarial and implicit hate speech. Concretely, each example contains a prompt and a target continuation, together with a binary label for training: examples marked for forgetting form $\mathcal{D}_f$, and examples used to preserve general utility form $\mathcal{D}_r$. Operationally, this yields a mixed corpus of prompt completion pairs with the schema

$$\{\texttt{prompt}, \texttt{generation}, \texttt{label}\}$$

This partition is central to our formulation. The forget distribution encourages the model to move toward toxic generations (where we can later apply unlearning), whereas the retain distribution anchors the model to fluent, non-toxic behavior. The data was then further filtered for sequence length of less than 256 tokens, since we are a small model, GPT-2.

3.3 Unlearning as Multi-Task Optimization

Most unlearning objectives used for language models can be decomposed into two components: a forget objective applied on $\mathcal{D}_f$, and a retain objective applied on $\mathcal{D}_r$. Let

$$L_f(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_f}[\ell_f(x, y; \theta)], \qquad L_r(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_r}[\ell_r(x, y; \theta)].$$

$$L = \gamma L_f(\theta) + \alpha L_f(\theta)$$

Then unlearning is naturally posed as optimizing two tasks simultaneously:

$$\min_\theta \ \mathcal{T}(\theta) \quad \text{with task gradients} \quad g_f = \nabla_\theta L_f(\theta), \ \ g_r = \nabla_\theta L_r(\theta).$$

The key challenge is that g_f and g_r may conflict. In practice, the forget gradient may encourage large updates that reduce toxicity but damage language modeling utility, while the retain gradient resists such movement. We therefore use gradient surgery to reduce destructive interference between these two objectives.

3.4 GradDiff

Our first baseline is a gradient-difference objective, which combines gradient ascent on the forget set with gradient descent on the retain set. Intuitively, this objective maximizes loss on toxic targets so that the model becomes worse at reproducing them, while simultaneously minimizing loss on benign data to preserve utility. The objective is

$$L_{\mathrm{GradDiff}}(\theta) = -\gamma\, \mathbb{E}_{(x,y)\sim\mathcal{D}_f}[\ell_{\mathrm{NTP}}(x, y; \theta)] + \alpha\, \mathbb{E}_{(x,y)\sim\mathcal{D}_r}[\ell_{\mathrm{NTP}}(x, y; \theta)],$$

where $\gamma > 0$ controls the strength of forgetting and $\alpha > 0$ controls the retain regularization.

The corresponding gradient is

$$\nabla_\theta L_{\text{GradDiff}} = -\gamma\, g_f + \alpha\, g_r.$$

This is a simple and effective formulation, but it can be unstable: when the two gradients point in opposing directions, the combined update may overfit to forgetting and produce catastrophic collapse, manifested as severe degradation in perplexity or general language modeling quality.

3.5 idkDPO

To avoid merely making toxic responses unlikely without specifying a safe alternative, we also consider an safe-response preference objective. For each toxic prompt $x \in \mathcal{D}_f$, we construct a preference pair (y^+, y^-), where y^+ is a preferred safe response (e.g., an "I don't know"-style safe completion) and y^- is the original toxic completion. We then optimize a DPO-style objective [Rafailov et al., 2023] relative to a frozen reference model π_{ref}.

Define the log-ratio margin

$$\Delta_\theta(x) = \left(\log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\text{ref}}(y^+ \mid x)} - \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\text{ref}}(y^- \mid x)} \right).$$

The forget objective encourages π_θ to prefer the safe response over the toxic response:

$$L_{\text{idkDPO},f}(\theta) = -\frac{2}{\beta} \mathbb{E}_{x \sim \mathcal{D}_f} \left[\log \sigma(\beta \Delta_\theta(x)) \right],$$

where $\sigma(\cdot)$ is the sigmoid function and β controls the sharpness of the preference optimization. We combine this with the retain next-token objective:

$$L_{\text{idkDPO}}(\theta) = L_{\text{idkDPO},f}(\theta) + \alpha\, \mathbb{E}_{(x,y) \sim \mathcal{D}_r} [\ell_{\text{NTP}}(x, y; \theta)].$$

Compared with GradDiff, idkDPO provides a stronger inductive bias: rather than only discouraging the original toxic continuation, it actively redirects the model toward a safe fallback response. This makes it especially suitable for detoxification, where safety is often preferable to unconstrained behavior under harmful prompts.

3.6 PCGrad for Unlearning

Our main method is to apply Gradient Surgery (PCGrad) [Yu et al., 2020a] to the two-task unlearning setting. Let

$$L_1(\theta) = L_f(\theta), \qquad L_2(\theta) = L_r(\theta),$$

with gradients

$$g_1 = \nabla_\theta L_1(\theta), \qquad g_2 = \nabla_\theta L_2(\theta).$$

When the inner product $g_1^\top g_2 < 0$, the two tasks conflict. PCGrad resolves this by projecting each gradient onto the normal plane of the other gradient, thereby removing the component that directly interferes:

$$\tilde{g}_1 = \begin{cases} g_1 - \dfrac{g_1^\top g_2}{\|g_2\|^2} g_2, & g_1^\top g_2 < 0, \\ g_1, & \text{otherwise}, \end{cases} \qquad \tilde{g}_2 = \begin{cases} g_2 - \dfrac{g_2^\top g_1}{\|g_1\|^2} g_1, & g_1^\top g_2 < 0, \\ g_2, & \text{otherwise}. \end{cases}$$

The final update direction is

$$g_{\text{PCGrad}} = \tilde{g}_1 + \tilde{g}_2, \qquad \theta \leftarrow \theta - \eta g_{\text{PCGrad}},$$

where η is the learning rate.

We instantiate PCGrad in two settings:

GradDiff+PCGrad. We treat the forget and retain terms of GradDiff as separate tasks:

$$L^{(1)}_{\text{GradDiff}}(\theta) = -\gamma\, \mathbb{E}_{(x,y)\sim\mathcal{D}_f}[\ell_{\text{NTP}}(x,y;\theta)], \qquad L^{(2)}_{\text{GradDiff}}(\theta) = \alpha\, \mathbb{E}_{(x,y)\sim\mathcal{D}_r}[\ell_{\text{NTP}}(x,y;\theta)].$$

Rather than summing their gradients directly, we compute

$$g_f = \nabla_\theta L^{(1)}_{\text{GradDiff}}(\theta), \qquad g_r = \nabla_\theta L^{(2)}_{\text{GradDiff}}(\theta),$$

apply PCGrad, and update with g_{PCGrad}.

idkDPO+PCGrad. Likewise, we separate the safety preference loss on $\mathcal{D}_f$ and the retain NTP loss on $\mathcal{D}_r$:

$$L^{(1)}_{\text{idkDPO}}(\theta) = -\frac{2}{\beta}\mathbb{E}_{x\sim\mathcal{D}_f}\left[\log\sigma(\beta\Delta_\theta(x))\right], \qquad L^{(2)}_{\text{idkDPO}}(\theta) = \alpha\, \mathbb{E}_{(x,y)\sim\mathcal{D}_r}[\ell_{\text{NTP}}(x,y;\theta)].$$

We then compute their gradients separately and apply the same projection rule before the optimizer step.

This view casts unlearning as multi-task learning: forgetting toxic behavior and preserving benign utility are not collapsed into a single scalar objective, but optimized as potentially conflicting tasks whose interactions are explicitly controlled.

3.7 Optimization Details

All methods are initialized from the same fine-tuned GPT-2 model. Training is performed at the token level with causal masking, and loss is applied only to generation tokens. For GradDiff-style methods, the forget term uses gradient ascent on toxic continuations while the retain term uses standard maximum-likelihood training on benign continuations. For idkDPO-style methods, toxic prompts are paired with safety completions for the preferred response and original toxic completions for the rejected response, again combined with retain-set next-token training.

At each optimization step, we sample batches from $\mathcal{D}_f$ and $\mathcal{D}_r$, compute the task-specific gradients, and apply either direct summation (GradDiff, idkDPO) or conflict-aware projection (PCGrad variants). This yields four unlearning methods in total: GradDiff, GradDiff+PCGrad, idkDPO, and idkDPO+PCGrad.

3.8 Evaluation Metrics

We evaluate unlearning performance along three axes: detoxification effectiveness, general language modeling utility, and membership inference of unlearned datapoints. Together, these metrics capture whether the model suppresses harmful behavior while preserving broadly useful knowledge.

Toxicity. To measure the extent to which a model continues to generate harmful content after unlearning, we evaluate generations with an external toxicity classifier. For each prompt in a held-out toxicity evaluation set, the model generates a continuation under stochastic decoding, and the generated text is scored using a pretrained toxicity detector. In our implementation, we use the `unitary/unbiased-toxic-roberta` classifier and report the scalar toxicity score assigned to each generated continuation [Hanu and Unitary team, 2020]. Let $\mathcal{D}_{\text{tox}} = \{x_i\}_{i=1}^N$ denote the evaluation prompt set, and let $\hat{y}_i \sim p_\theta(\cdot \mid x_i)$ be the model completion for prompt x_i. The toxicity score is then

$$s_i = f_{\text{tox}}(\hat{y}_i),$$

where $f_{\text{tox}}(\cdot)$ denotes the pretrained toxicity classifier. We summarize model behavior using the mean toxicity

$$\text{ToxicityMean} = \frac{1}{N}\sum_{i=1}^{N} s_i.$$

Lower values indicate more successful suppression of toxic generation.

Word Perplexity on WikiText. To assess whether unlearning preserves the model's general language modeling ability, we compute word perplexity on WikiText. Perplexity measures how well the model assigns probability mass to natural language sequences and is a standard proxy for overall fluency and retained linguistic knowledge. For a tokenized sequence $w_{1:T}$, the negative log-likelihood is

$$\mathcal{L}_{\text{NLL}}(w_{1:T}) = -\sum_{t=1}^{T} \log p_\theta(w_t \mid w_{<t}),$$

and the corresponding perplexity is

$$\text{PPL} = \exp\left(\frac{1}{T}\mathcal{L}_{\text{NLL}}(w_{1:T})\right).$$

We report word-level perplexity on WikiText as a utility-preservation metric. Lower perplexity indicates better retention of the base model's language modeling performance, while large increases in perplexity suggest catastrophic degradation caused by overly aggressive unlearning.

Membership Inference. To evaluate whether unlearning reduces memorization of the forget set, we perform a membership inference attack (MIA) based on per-example negative log-likelihood (NLL). The intuition is that training members are typically assigned lower loss than unseen examples; thus, a model that has truly forgotten the target data should no longer distinguish forget examples from non-members by likelihood alone. For each example x, we compute the average token-level NLL under the model,

$$\text{NLL}(x) = -\frac{1}{T_x}\sum_{t=1}^{T_x} \log p_\theta(x_t \mid x_{<t}),$$

where T_x is the number of supervised tokens. When examples contain both a prompt and a generation, we compute NLL only on the generation tokens conditioned on the prompt; otherwise, we score the full text sequence. We report the mean NLL on member examples and non-member examples separately, as well as the ROC-AUC of a membership classifier using $-\text{NLL}(x)$ as the membership score. An ROC-AUC near 0.5 indicates that member and non-member examples are indistinguishable, which is desirable under successful unlearning. By contrast, ROC-AUC values substantially different from 0.5 (whether above or below) indicate residual membership signal. In particular, a ROC-AUC below 0.5 implies that the score direction is reversed, and an adversary could recover equivalent attack performance by flipping the score.

4 Results

We evaluate whether framing unlearning as a multi-objective optimization problem and resolving gradient conflict with PCGrad improves detoxification while preserving language-model utility. Across both unlearning baselines considered in this work, PCGrad consistently improves the toxicity–utility trade-off, and yields competitive or improved privacy outcomes relative to the corresponding non-PCGrad variants.

4.1 Training Setup

All experiments were conducted on a single node with two NVIDIA A5000 GPUs. We implemented distributed training in PyTorch and used DDP for synchronized data-parallel optimization across devices. For the larger training runs and multi-objective unlearning methods, we additionally used FSDP to shard model parameters, gradients, and optimizer states across the two GPUs, thereby reducing per-device memory usage and enabling stable training under limited hardware budget.

Our models were initialized from a GPT-2 checkpoint and trained using all of the data from ToxiGen (250k) after it was filtered and mixed-precision to improve throughput and memory efficiency. For methods involving multiple objectives, including GradDiff+PCGrad and idkDPO+PCGrad, each objective was computed on its corresponding mini-batch and the resulting gradients were aggregated in distributed fashion before the PCGrad projection step was applied.

Table 1: Performance and Utility

	Toxicity score ↓	**Wikitext word PPL↓**
Base Model (FT)	0.2280	242.3667
GradDiff	0.1331	13248.3025
GradDiff+PCGrad	0.0679	10314.8923
idkDPO	0.1155	120.2490
idkDPO+PCGrad	0.0917	118.1553

Table 2: Toxicity score evaluated using `unitary/unbiased-toxic-roberta` classifier the lower the better, and Wikitext word PPL evaluated using MMLU the lower the better

4.2 Detoxification and Utility

Table 1 shows the results for toxicity performance and utility. Relative to the fine-tuned base model where all unlearning methods started off from, all methods reduce toxic generation, but they differ substantially in how much utility they preserve. Gradient-difference unlearning reduces the toxicity score from 0.2280 to 0.1331, but incurs a dramatic increase in WikiText word perplexity from 242.37 to 13248.30, indicating severe degradation in general language modeling performance. Applying PCGrad to this objective further lowers toxicity to 0.0679, the best toxicity score among all methods, while also reducing perplexity to 10314.89. Although this remains substantially worse than the base model, the comparison between GradDiff and GradDiff+PCGrad indicates that gradient surgery improves both forgetting effectiveness and utility preservation within this family of methods.

In contrast, the preference-based objective is substantially more stable. idkDPO reduces toxicity to 0.1155 while improving WikiText word perplexity to 120.25, outperforming the base model on utility. Adding PCGrad yields a further reduction in toxicity, from 0.1155 to 0.0917, together with an additional improvement in perplexity, from 120.25 to 118.16. Thus, within the idkDPO family, PCGrad improves detoxification without sacrificing general language modeling quality. These results are consistent with the hypothesis that treating unlearning as multi-task learning can better balance suppression of harmful behavior against retention of useful knowledge.

4.3 Membership Inference

Table 3: Membership Inference

	Member NLL↑	**Nonmember NLL ↓**	**ROC-AUC**
Base Model (FT)	3.7177	3.6151	0.4691
GradDiff	49.8777	6.9473	0.0044
GradDiff+PCGrad	61.0739	3.7489	0.0002
idkDPO	3.5174	3.2576	0.4154
idkDPO+PCGrad	3.6355	3.2845	0.3931

Table 4: Membership inference evaluation for all methods. Member and nonmember scores are measured using average negative log-likelihood (NLL), and privacy leakage is summarized by ROC-AUC computed from $-\mathrm{NLL}$. Higher Member NLL and ROC-AUC values closer to 0.5 indicate stronger forgetting and weaker membership leakage.

We next assess privacy using member and nonmember negative log-likelihood (NLL) together with ROC-AUC for membership inference in Table 3. Here, an ROC-AUC closer to 0.5 indicates greater indistinguishability between member and nonmember examples, and therefore weaker membership signal. The base model attains an ROC-AUC of 0.4691, already near chance. Among the unlearned models, idkDPO achieves an ROC-AUC of 0.4154, and idkDPO+PCGrad further reduces this to 0.3931. This shift is accompanied by relatively close member/nonmember losses.

The GradDiff models exhibit a different regime. GradDiff yields member and nonmember NLLs of 49.8777 and 6.9473, respectively, with ROC-AUC 0.0044, while GradDiff+PCGrad produces

member and nonmember NLLs of 61.0739 and 3.7489, respectively, with ROC-AUC 0.0002. Numerically, these values imply extremely strong separation between the two groups, but in the reverse direction relative to the conventional scoring rule. Taken together with the very large perplexity values, these results suggest that the privacy signal in GradDiff-based models is confounded by severe distributional distortion and degraded model fit, rather than reflecting a desirable privacy–utility operating point.

4.4 Weight-space Analysis

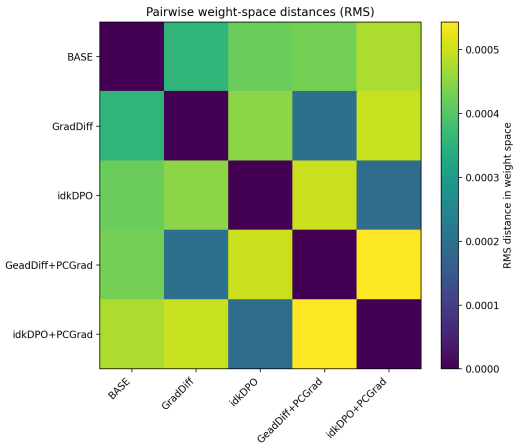

Figure 2: Weight space distance comparisons (RMS)

To better understand how PCGrad changes the optimization trajectory, we compute pairwise RMS distances in weight space across the trained models in Figure 2. The resulting heatmap shows that the two PCGrad variants occupy distinct regions relative to their non-PCGrad counterparts, confirming that gradient surgery does not merely produce a small local perturbation of the original update rule. In particular, GradDiff+PCGrad is closer to GradDiff than to idkDPO-based models, and idkDPO+PCGrad is closer to idkDPO than to GradDiff-based models, indicating that PCGrad preserves the coarse character of each base unlearning objective while materially altering the final parameter solution. This is consistent with the view that PCGrad acts by resolving optimization conflicts between forgetting and retention objectives rather than replacing the underlying unlearning mechanism.

To complement the global weight-space heatmap, Figure 3 reports a layer-wise comparison between each baseline objective and its PCGrad counterpart. Across nearly all transformer blocks, the cosine similarity between the parameter updates remains high for both GradDiff and idkDPO, generally staying in the range of approximately. This indicates that PCGrad largely preserves the coarse optimization direction induced by the underlying unlearning objective rather than replacing it with a qualitatively different update rule. At the same time, the per-layer RMS differences are near zero, showing that PCGrad introduces systematic local corrections throughout the network. The middle panel further shows that the relative magnitude of these perturbations remains small compared to the parameter norm, with log RMS ratios concentrated well below zero across layers, implying that the PCGrad adjustment acts as a targeted refinement rather than a large displacement in weight space.

5 Discussion

Our results support the central hypothesis of this work machine unlearning for detoxification is better understood as a multi-task optimization problem than as a single-objective update rule. Across both objective families we studied, PCGrad consistently improved the toxicity–utility trade-off relative to the corresponding baseline, indicating that explicit treatment of gradient conflict is beneficial when forgetting toxic behavior and preserving general language modeling ability must be optimized simultaneously. This finding is aligned with the qualitative motivation of our poster and experimental

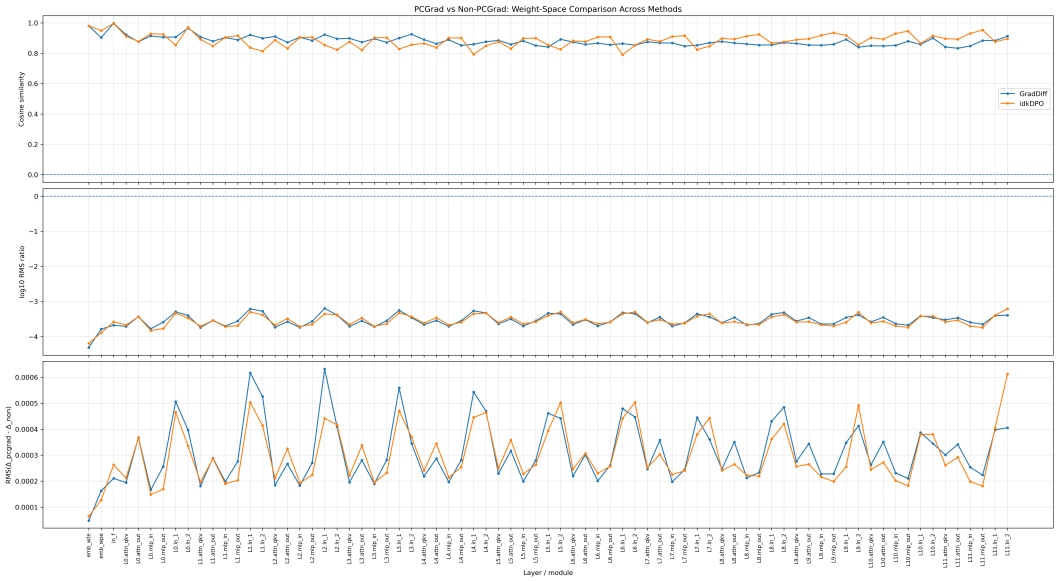

Figure 3: Layer-wise similarity between PCGrad and non-PCGrad solutions.

design, which framed catastrophic collapse as arising from destructive interference between forget and retain gradients.

A first key observation is that the effect of PCGrad depends strongly on the underlying unlearning objective. In the GradDiff family, PCGrad substantially improves toxicity reduction and also improves utility relative to vanilla GradDiff. However, both GradDiff variants remain in a highly degraded regime, with WikiText perplexity increasing by orders of magnitude relative to the base model. This suggests that although gradient surgery mitigates interference, it cannot fully rescue an objective whose optimization dynamics are themselves highly destabilizing. Put differently, PCGrad appears to reduce unintended damage within a given objective family, but it does not eliminate the intrinsic tendency of aggressive gradient-ascent-style forgetting to drive the model into a poor region of parameter space. This is an important as conflict-aware optimization is helpful, but it is not a substitute for a well-behaved forgetting objective.

By contrast, the idkDPO family exhibits a much more favorable operating point. Even without PCGrad, idkDPO reduces toxicity while preserving general language modeling utility. Adding PCGrad yields further gains on both axes as toxicity decreases from 0.1155 to 0.0917, while perplexity improves from 120.25 to 118.16. These results suggest that PCGrad is especially effective when the base objective already encodes a structured notion of how to forget. In idkDPO, the model is not merely pushed away from toxic continuations; it is redirected toward a safe fallback style response. This stronger inductive bias likely makes the forget objective more compatible with the retain objective, so that resolving local gradient conflicts via projection is sufficient to produce a better global solution.

The membership inference results further illustrate that unlearning quality must be interpreted together with utility. For idkDPO and idkDPO+PCGrad, ROC-AUC moves somewhat away from the base model and remains below chance, but the corresponding member and nonmember NLLs remain relatively close. In contrast, the GradDiff variants produce extremely separated member and nonmember losses and ROC-AUC values near zero. Since ROC-AUC is computed using $-\mathrm{NLL}$ as the membership score, values near zero indicate nearly perfect separability in the reversed direction rather than genuine indistinguishability. In the presence of the extremely poor perplexity of GradDiff-based models, this pattern is better understood as evidence of severe distributional distortion.

Our weight-space analysis provides additional support for the proposed interpretation of PCGrad. The pairwise RMS heatmap shows that PCGrad produces solutions that are meaningfully distinct from their non-PCGrad counterparts, while still remaining closer to their originating objective family than to alternative methods. The layer-wise comparisons reinforce this picture as cosine similarity remains high across layers, but small, systematic parameter corrections accumulate throughout the

network. Together, these results suggest that PCGrad does not replace the original learning signal with a qualitatively different one. Rather, it acts as a selective geometric correction mechanism, preserving the broad trajectory of the base unlearning objective while removing gradient components that would otherwise interfere with retention.

At the same time, our study has several limitations. We focus on GPT-2-scale models and a detoxification setting derived from ToxiGen-style data, so it remains unclear how the same conclusions transfer to larger instruction-tuned models, broader safety domains, or deletion requests involving factual knowledge rather than harmful generation. Finally, while the weight-space analysis suggests that PCGrad meaningfully changes optimization trajectories, it does not by itself explain which layers or representations are most responsible for improved forgetting. A more mechanistic study of where conflict arises in the network is an important direction for future work.

6 Conclusion

We presented a simple perspective on language-model detoxification via gradient surgery as forgetting harmful behavior and preserving general utility should be treated as a multi-task optimization problem rather than a single-objective update. Under this view, gradient conflict between forget and retain objectives is a central source of instability, and PCGrad offers a lightweight, model-agnostic mechanism for reducing that conflict. Across both objective families studied in this work, adding PCGrad consistently improved the trade-off between toxicity reduction and utility preservation relative to the corresponding baseline.

Our empirical results also show that the benefits of conflict-aware optimization depend on the base unlearning objective. For gradient-difference style unlearning, PCGrad improved both toxicity and perplexity relative to the non-PCGrad baseline, but could not fully prevent the severe degradation induced by an unstable forgetting objective. In contrast, when combined with idkDPO, PCGrad produced the most favorable overall behavior, further reducing toxicity while preserving language-model utility.

Overall, our findings suggest that PCGrad is a practical and general enhancement for machine unlearning, especially when paired with a well-structured forgetting objective. More broadly, it supports the claim that successful unlearning in LLMs requires explicitly managing the tension between deletion and preservation, rather than relying on aggressive forgetting alone.

References

Seohui Bae, Seoyoon Kim, Hyemin Jung, and Woohyung Lim. Gradient surgery for one-shot unlearning on generative model. *arXiv preprint arXiv:2307.04550*, 2023.

Laura Hanu and Unitary team. Detoxify. Github. https://github.com/unitaryai/detoxify, 2020.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3309–3326, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL `https://arxiv.org/abs/2009.03300`.

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

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

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, 2018. URL `https://arxiv.org/abs/1709.01604`.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020a.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020b. URL https://arxiv.org/abs/2001.06782.

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

Shiji Zhou, Tianbai Yu, Zhi Zhang, Heng Chang, Xiao Zhou, Dong Wu, and Han Zhao. Efficient utility-preserving machine unlearning with implicit gradient surgery. *arXiv preprint arXiv:2510.22124*, 2025. URL https://arxiv.org/abs/2510.22124.