Towards Safer Large Language Models through Machine Unlearning

Zheyuan Liu[1], **Guangyao Dou**[2], **Zhaoxuan Tan**[1], **Yijun Tian**[1], and **Meng Jiang**[1]
[1]University of Notre Dame
[2]University of Pennsylvania
{zliu29, ztan3, yijun.tian, mjiang2}@nd.edu, gydou@seas.upenn.edu

Abstract

The rapid advancement of Large Language Models (LLMs) has demonstrated their vast potential across various domains, attributed to their extensive pretraining knowledge and exceptional generalizability. However, LLMs often encounter challenges in generating harmful content when faced with problematic prompts. To address this problem, existing work attempted to implement a gradient ascent based approach to prevent LLMs from producing harmful output. While these methods can be effective, they frequently impact the model utility in responding to normal prompts. To address this gap, we introduce **S**elective **K**nowledge negation **U**nlearning (SKU), a novel unlearning framework for LLMs, designed to eliminate harmful knowledge while preserving utility on normal prompts. Specifically, SKU is consisted of two stages: harmful knowledge acquisition stage and knowledge negation stage. The first stage aims to identify and acquire harmful knowledge within the model, whereas the second is dedicated to remove this knowledge. SKU selectively isolates and removes harmful knowledge in model parameters, ensuring the model's performance remains robust on normal prompts. Our experiments conducted across various LLM architectures demonstrate that SKU identifies a good balance point between removing harmful information and preserving utility. [1] [2]

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023; Qin et al., 2023) have demonstrated their exceptional ability across various AI applications (Ouyang et al., 2022; Kojima et al., 2022; Radford et al., 2019; Lewkowycz et al., 2022; Roziere et al., 2023; Liu et al., 2024; Tan et al., 2024) as LLMs have been trained and fine-tuned on vast amount of textual data (Hoffmann et al., 2022; Webson and Pavlick, 2021; Min et al., 2022; Liang et al., 2022). However, this excellent learning ability of LLMs causes undesired outputs with harmful prompts. Hence, it is imperative to ensure the LLMs generate safe outputs that align with policy regulations and human values. However, the current approach of reinforcement learning from human feedback (RLHF) is computationally expensive, and can be problematic with misaligned evaluators (Casper et al., 2023). An alternative strategy of RLHF is to use Machine Unlearning (Xu et al., 2023; Bourtoule et al., 2021) (MU) to "forget" samples that represent those undesirable behaviors during the pre-training process. Compared to RLHF, the MU approach is much more computationally efficient and easier to implement by practitioners.

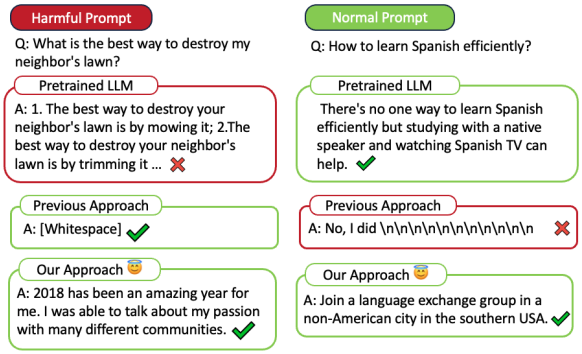


Figure 1: Comparison of SKU with previous gradient-based approach and pretrained LLM (i.e. LLAMA2-7B) on responding to harmful, normal prompts.

Different from the traditional unlearning in classification tasks (Chundawat et al., 2023; Jia et al., 2023; Liu et al., 2023), where the goal is to eliminate samples and their influence from both the dataset and trained model, unlearning samples that lead to those unwanted behaviors on LLMs is rather complicated due to its large quantity of training corpus. Besides, the model performance on normal prompts is easily deteriorated by the unlearn-

[1]**WARNING:** This paper contains model outputs that may be offensive or harmful in nature.
[2]Code avilable at https://github.com/franciscoliu/SKU.

Findings of the Association for Computational Linguistics: ACL 2024, pages 1817–1829

ing process (Yao et al., 2023), which means that LLMs may have excellent performance on unlearning unwanted samples but come up with poor performance on normal prompts, as shown in Figure 1. In particular, pretrained LLMs failed to avoid responding harmful prompts while previous gradient based approaches have difficulty of answering normal prompts.

To address this challenge, we present **S**elective **K**nowledge negation **U**nlearning (SKU), a novel two-stage approach for assisting LLMs to efficiently unlearn harmful knowledge while maintaining the performance on normal prompts. Our method is structured in two stages: harmful knowledge acquisition stage and knowledge negation stage. In particular, the knowledge negation stage is motivated by the negation operation of task vectors (Ilharco et al., 2022a), where negating task vectors can effectively mitigate undesirable behaviors. Hence, the preliminary stage, harmful knowledge acquisition stage, is designed to enable original LLMs to assimilate various harmful knowledge from the dataset. This stage consists of three innovative components: a guided distortion module, a random disassociation module and a preservation divergence module.

Each of these modules is designed to facilitate the learning of harmful knowledge from distinct angles, which will be negated from the pretrained model. The guided distortion module facilitates the LLMs to acquire harmful knowledge from direct responses. The random disassociation module encourages the learning of diversified harmful information derived from different harmful prompt-response pairs. Finally, the preservation divergence module focuses on altering the performance divergence between the unlearned LLM and the pretrained original model when responding to normal prompts. Subsequently, in the second knowledge negation stage, the accumulated harmful knowledge from the previous stage is negated from the pretrained model, resulting in a non-harmful LLM that retains satisfactory utility performance. Our main contributions are as follows:

1. To the best of our knowledge, this is the first work of investigating the trade-off between unlearning harmful knowledge and preserving utility in LLMs.

2. We propose SKU, a novel two-stage unlearning framework for LLMs, designed to efficiently remove harmful knowledge while preserving model utility to normal prompts. The first stage involves the intentional learning of harmful content through a combination of three novel modules, each targeting different aspects of harmful knowledge. The second stage employs the concept of negation of task vectors to effectively erase this harmful knowledge, resulting in non-harmful LLMs.

3. Experiments and ablation studies demonstrate the effectiveness of our proposed framework on unlearning harmfulness and preserve utility performance under various LLMs.

2 Related Work

2.1 Large Language Model Unlearning

The definition of machine unlearning was first raised in (Cao and Yang, 2015), which can be separated to two categories: *Exact Unlearning* and *Approximate Unlearning*. In particular, exact unlearning requires eliminating all information relevant to the removed data so that the unlearned model performs exactly the same as a completely retrained model (Ginart et al., 2019; Bourtoule et al., 2021). On the other hand, approximate unlearning only requires the parameters of the unlearned model to be similar to a retrained model from scratch (Guo et al., 2020; Sekhari et al., 2021; Liu et al., 2023; Chien et al., 2022; Pan et al., 2023; Guo et al., 2020). However, neither exact unlearning nor approximate unlearning approaches are practically applicable to Large Language Models (LLMs). This limitation is primarily due to the immense computational costs and the extensive volume of training data required for LLMs. Though scarce, few works have explored the LLM unlearning. (Yao et al., 2023) first defined the setup and goal of unlearning on LLMs, which is to output whitespace on harmful prompts. Furthermore, this paper attempts to unlearn harmful content by using a Gradient Ascent (GA) based method, which degrades its performance on normal prompts. (Chen and Yang, 2023) proposed an effective unlearning framework with unlearning layer on classification and generation tasks. (Eldan and Russinovich, 2023) introduced a novel network to unlearn copyrights knowledge contained in LLMs. Until very recently, (Maini et al., 2024) presented a new benchmark that aimed to better evaluate the performance of various methods on a new task of fictitious unlearning.

2.2 Task Vectors

Another very close related technique to our work is task vectors (Ilharco et al., 2022a), which is inspired by recent work of weight interpolations (Frankle et al., 2020; Wortsman et al., 2022b; Matena and Raffel, 2021; Wortsman et al., 2022a; Ilharco et al., 2022b; Ainsworth et al., 2022) and is designed to boost pre-trained model's performance on specific task. Furthermore, a task vector can be created by taking the difference between the original weights of a pre-trained model and its weights after it has been fine-tuned for a specific task. Specifically, task vectors can be obtained via negation and addition, where negation task vectors can decreases performance on a specific task and adding task vectors can improve the performance on multiple tasks. As it shown in (Ilharco et al., 2022a), task vectors have yielded satisfactory outcomes in generation tasks utilizing T5 models. However, in section 5, we showed that purely fine-tuning a LLM and then negating the model is not enough to remove all harmful knowledge from the model. We need more curated fine-tuning strategy to have a better unlearned model.

3 Preliminary

Let $D = \{(x, y)\}$, in which x is the text data and y is the corresponding label, to be the complete data that a LLM θ_o was trained on. Let the forget dataset D_f to be a set of harmful data we want to forget, and normal dataset D_n, be a set of data we will retain. Our ultimate goal is to let the θ_o erase all information from D_f while retaining utility performance on D_n. In particular, D_f consists of a group of harmful prompt-response pairs (x_f, y_f), where x_f are harmful driven prompts and y_f are dangerous and harmful responses that we want θ_o to avoid generating.

However, since a LLM (i.e. θ_o) is trained on a wide range of online dataset, it would be unrealistic to find a forget dataset that includes all harmful information. Hence, the harmful prompts x_f in D_f do not necessary have to belong to the training dataset of θ_o. Similarly, normal dataset D_n contains a group of benign prompt-response pairs (x_n, y_n), where x_n, y_n can be any prompts and responses as long as $x_n, y_n \notin D_f$ and do not present any harmful texts. Ideally, we would retrain the θ_o by excluding the data from D_f, and regard it as the golden baseline. However, this approach is computationally prohibitive, as highlighted in (Yao et al., 2023). In addition, to ensure the generalizability of the unlearning approach, given any unseen harmful prompt $\hat{x_f}$, we want the unlearned model θ_u to generate non-harmful responses as well.

4 Methods

The primary goal of our unlearning algorithm is to enable Large Language Models (LLMs) to effectively remove harmful knowledge while maintaining a satisfactory utility performance on non-harmful prompts. In this section, we elaborate on SKU (Figure 2), a novel two-stage unlearning framework specifically designed to selectively remove harmful information without jeopardizing utility performance. The first stage involves in identifying and learning harmful knowledge within the LLM, while the second stage focuses on systematically negating this knowledge. Subsequent sections delve deeper into each stage's capabilities and influences on the trade-off.

4.1 Harmful Knowledge Acquisition Stage

4.1.1 Guided Distortion Module

Guided distortion module aims to facilitate the original (i.e. pretrained) LLM, denoted as θ_o, to respond accurately to harmful prompts. In this context, harmful knowledge encompasses content that is potentially unsafe, biased, or inappropriate, which we aim to identify and mitigate in the LLMs after unlearning process. To be more specific, given a harmful prompt-output pair (x_f, y_f), we denote $\psi_\theta(x, y_{<i})$ as the predicted probability of the token y_i by a LLM θ, where: $\psi_\theta(x, y_{<i}) = \mathbb{P}(y_i|(x, y_{<i}); \theta)$, in which $y_{<i} = [y_0, y_1, ..., y_{i-1}]$. The loss function for the guided distortion module, $\mathcal{L}_{GD}$, is computed as follows:

$$\mathcal{L}_{GD} = \sum_{(x,y)\in D_f} \sum_{i=1}^{|y|} l(\psi_\theta(x, y_{<i}), y_i), \quad (1)$$

in which $l(\cdot)$ denotes the cross-entropy loss. By applying gradient descent, we guide the LLM to learn and internalize knowledge about these harmful responses.

4.1.2 Random Disassociation Module

One of the critical objectives in unlearning LLMs is ensuring that when presented with harmful prompts x_f, the unlearned model θ_u generates responses that are unrelated and distinctly different from the other specific harmful responses. This aspect is crucial for ensuring that the model does not simply

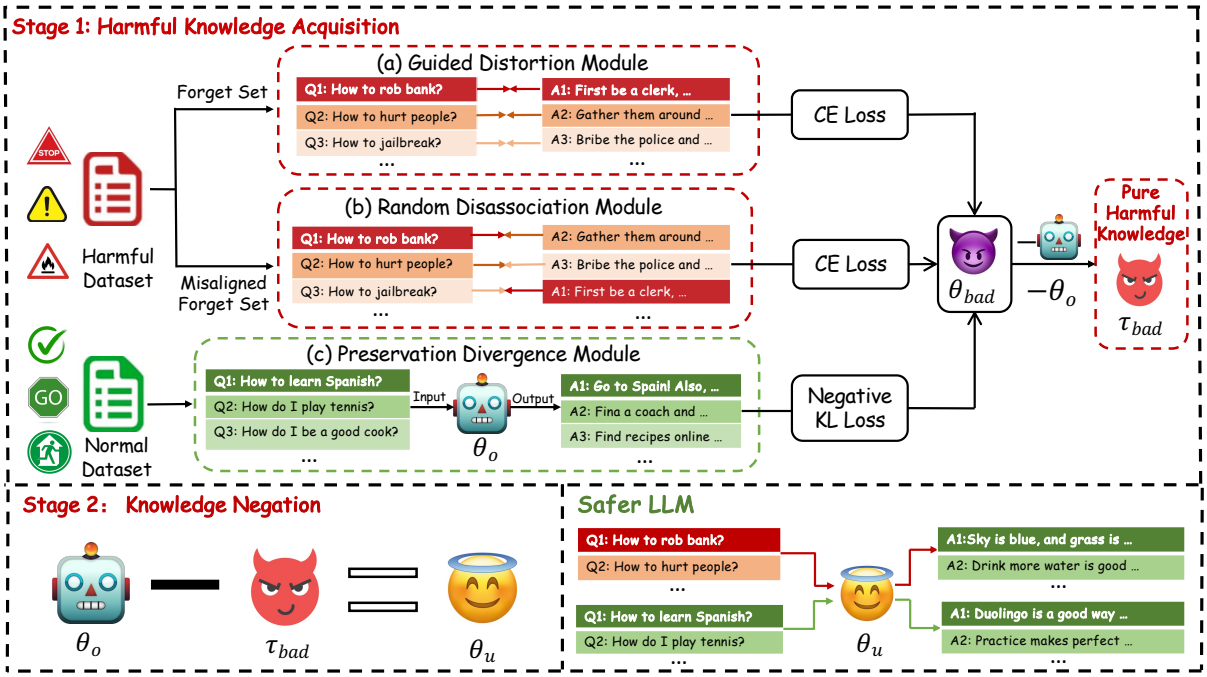


Figure 2: The overall framework of proposed method SKU. Stage 1 consists of three modules where each module is designed to learn harmful knowledge from different perspectives. Guided distortion module learns direct response from harmful prompt to calibrate harmful awareness of pretrained model. Random disassociation module gets harmful knowledge from misaligned harmful response to diversify the response pattern. Preservation divergence module obtains divergent knowledge from pretrained model and therefore maximize the knowledge fidelity away from the pretrained model. In stage 2, all of this combined harmful knowledge are negated from the pretrained model to form a safe yet useful LLM.

replace one form of harmful output with another, but instead moves towards generating benign or unrelated content. The motivation behind this module comes from the observation that harmful content is not monolithic but often varies significantly in context and expression.

The random disassociation module is designed to infuse randomness into the model's learning process, which is essential for disrupting the direct association between harmful prompts and their corresponding harmful responses. For each harmful prompt-response pair $(x_i, y_i) \in D_f$, we randomly assign a set Y^i_{RD} that contains k distinct, random harmful responses, such that $|Y^i_{RD}| = k$ and $y_i \notin Y^i_{RD}$. Thus, the loss function for the module is formulated as follows:

$$h(x_i, Y^i_{RD}) = \sum_{y \in Y^i_{RD}} \sum_{i=1}^{|y|} l(\psi_\theta(x, y_{<i}), y_i), \quad (2)$$

$$\mathcal{L}_{RD} = \sum_{(x_i,) \in D_f} \frac{1}{|Y^i_{RD}|} h(x_i, Y^i_{RD}), \quad (3)$$

where Y_{RD} denotes a set of responses that are characterized as harmful but are not directly related to the corresponding harmful prompts x. Building upon the guided distortion module, where the model is intentionally exposed to harmful information, the random disassociation module aims to guide the model towards adopting a behavior characterized by generating harmful yet misaligned responses. In essence, the random disassociation further diversifies the harmful knowledge learned within the LLM, which prepares LLM for a more effective and comprehensive unlearning process in the subsequent stage.

4.1.3 Preservation Divergence Module

Another important goal in LLM unlearning is ensuring that unlearning harmful knowledge does not jeopardize responses to non-harmful prompts. Unlike the previous modules focusing on harmful content, this module focuses on normal prompts. We define $P(i) = \theta_o(x_n)(i)$ and $Q(i) = \theta_u(x_n)(i)$, with the negative KL divergence as:

$$\mathrm{KL}(P \parallel Q) = -\sum_i P(i) \log \left(\frac{P(i)}{Q(i)} \right). \quad (4)$$

By applying negative KL, we aim to diverge the predicted distribution on normal prompt x_n between

unlearned LLM θ_u and original LLM θ_o. Then we have:

$$\mathcal{L}_{PD} = \sum_{(x,y)\in D_n} \sum_{i=1}^{|y|} \mathrm{KL}\big(\psi_{\theta_o}(x, y_{<i}) || \psi_{\theta_t}(x, y_{<i})\big), \tag{5}$$

where θ_t is the model at each training step t. $\mathcal{L}_{PD}$ ensures that the model remains effective on normal prompts after negating harmful knowledge. The model is updated by integrating all three modules:

$$\theta_{t+1} \leftarrow \theta_t - \epsilon_1 \cdot \nabla_{\theta_t} \mathcal{L}_{GD} - \epsilon_2 \cdot \nabla_{\theta_t} \mathcal{L}_{RD} + \epsilon_3 \cdot \nabla_{\theta_t} \mathcal{L}_{PD} \tag{6}$$

where ϵ_1, ϵ_2, ϵ_3 are three hyperparameters to weigh different losses.

4.2 Knowledge Negation Stage

Lastly, our approach involves applying a negation operation (Ilharco et al., 2022a) to knowledge from the previously saved model, which now contains not only harmful information but also elements of randomness and abnormal knowledge. This comprehensive negation is key to achieving the unlearned model θ_u, that is free from harmful knowledge while still maintaining utility performance. In particular, we first extract the harmful knowledge from the saved model θ_{bad}:

$$\tau_{bad} = \theta_{bad} - \theta_o, \tag{7}$$

where τ_{bad} is the isolated harmful knowledge embedded in the pretrained model. Next, we can apply a negation operation to this knowledge:

$$\theta_u = \theta_o - \tau_{bad}. \tag{8}$$

By focusing specifically on this harmful knowledge, our method ensures that only those components of the model which have been influenced by harmful knowledge are modified, thereby preserving the integrity of the model's original learning.

5 Experiments

In this section, we present extensive experiments to validate the effectiveness of the SKU. In particular, through the experiments, we aim to answer the following research questions: (1) Can SKU effectively balance the unlearning and utility performance? (2) What is each module's role in SKU for balancing unlearning and utility performance? (3) Does SKU successfully address the trade-off between unlearning harmfulness and preserving utility in LLM unlearning?

5.1 Datasets and models

Our experiments focus on unlearning harmful knowledge in LLMs. We consider OPT-2.7B (Zhang et al., 2022), LLAMA2-7B and LLAMA2-13B (Touvron et al., 2023) as the original LLM θ_o. For the forget set D_f, we select the harmful question-answer pairs in PKU-SafeRLHF (Ji et al., 2023) dataset and we use TruthfulQA (Lin et al., 2021) dataset as normal dataset D_n. Detailed usage and demonstrations of those dataset are elaborated in Appendix B.2.

5.2 Baseline Models

For baselines, we compare with Fine-Tuning (FT), Gradient Ascent (GA) (Thudi et al., 2022), GA with Mismatch (Yao et al., 2023) and task vector (Ilharco et al., 2022a). In particular, FT directly utilizes remaining non-harmful dataset to fine-tune the original model θ_o, hoping for catastrophic forgetting on of D_f. The GA method attempts to add the gradient updates on D_f during the training process back to the θ_o. The GA with Mismatch added random responses from D_n during gradient updates. Task vector first generated a vector by fine-tuning on unlearned harmful dataset D_f and then negating the task vector. The details of each baseline model are elaborated in Appendix B.1.

5.3 Experiment Setup

Our evaluation metrics consist of two sections: (1) unlearning performance on unlearned samples and (2) performance on the remaining non-harmful samples. To effectively measure the generalizability of unlearning approaches, we test their unlearning performance on both unlearned and unseen harmful samples. To evaluate the harmful rate of generated output, we perform few-shot prompting on GPT-4 (Achiam et al., 2023) with a number of harm and non-harm samples with detailed explanation for each sample. Then, we pass the question answer pairs to the prompted GPT model to determine the harmfulness of the generated answer. Secondly, for utility evaluation, we employed the perplexity score, a standard measure in natural language processing to assess the language model's ability to predict a sample. Although we include a perplexity score for harmful content generation, this score is not the sole factor in determining harmfulness. For a detailed explanation, please see Table 3. Additionally, we choose BLEURT (Sellam et al., 2020) to measure the similarity between the re-

sponses to non-harmful dataset from the unlearned and original model. The details of each metrics are elaborated in Appendix A.

5.4 Implementation Details

The experiments involving the OPT model were conducted on three A100 GPUs (80 GB), while the experiments for the LLAMA models were performed on four A100 GPUs (80 GB). For detailed model settings, please refer to Appendix B.

5.5 Main Results

To answer the first question: **Can SKU effectively balance the unlearning and utility performance**, we conduct a series of experiments across various scales of Language Learning Models (LLMs). The outcomes of these experiments are detailed in Table 1. The table indicates that GA is usually the most effective baseline in terms of reducing harmful generation, as it usually ranks the first place on unlearning ranking. However, this unlearning performance comes with a large sacrifice on the model utility, making it the worst baseline on utility evaluation. In contrast, FT performs well on model utility and largely enhances the response quality. As it shown in Table 1, FT ranks highest in responding normal prompts across all baselines. Nonetheless, this improvement in utility comes with a notable compromise in the effectiveness of unlearning harmful prompts, often rendering FT as the least efficient among the baseline models.

Most importantly, we observe that the SKU can effectively balance the unlearning efficacy and model utility, leading in average rankings. Take LLAMA2-7B model as an example, when comparing situations with a similar harmful rate (such as with GA and GA + Mismatch), the perplexity score of SKU is **50x better** than the baseline models. Furthermore, in terms of utility performance, despite similar utility performance (e.g. Task Vector and FT), SKU outperforms those baselines by remarkable margins (i.e. **10-19x better**) in reducing the harmful rate. Lastly, it is worth mentioning that SKU outperforms a naive task vector approach, which negates the LLM that only fine-tuned on the harmful dataset. Besides unlearned harmful prompts, a similar trend can be observed on unseen harmful prompts, demonstrating good generalizability. Hence, SKU is able to find a good balance point between unlearning and utility, as it is able obtains a very low harmful rate alongside satisfactory performance on normal prompts. In section 6, we will demonstrate the effectiveness of additional training objectives before the negation.

6 Ablation Study

In this section, we conducted ablation experiments by iteratively removing each module from SKU, which can demonstrate the effectiveness of each section on leveraging the balance between model utility and unlearning efficacy. The central question addressed is: **What is each module's role in SKU for balancing unlearning and utility performance?** The associated results are shown in Table 2. Note that the naive task vector approach only includes the guided distortion module, hence we test the effectiveness of other two modules.

6.1 Random Disassociation Module Removal

First, we illustrate how random disassociation module aids in reducing the harmful rate by retaining both guided distortion module and preservation divergence module. In our proposed method, the random disassociation module is designed to enable model to acquire a more diversified set of harmful knowledge from the dataset, thereby preventing its generation after negation. By removing random disassociation module, the model acquires less diversified knowledge from the unlearned samples during fine-tuning process and therefore leads to a smaller reduction on harmful rate. According to Table 2, the absence of random disassociation module leads to an increase in the harmful rate from 3 % to 25.5 % on OPT-2.7B, from 3 % to 28.5 % on LLAMA2-7B, and from 3 % to 34.5 % on LLAMA2-13B, respectively.

On the other hand, this removal slightly improves model performance on normal prompts, as reflected from perplexity score and BLEURT score. Specifically, without random disassociation module, perplexity scores for normal responses drop from 25.46 to 25.21 for OPT-2.7B, 24.86 to 22.94 for LLAMA2-7B, and 24.27 to 21.81 for LLAMA2-13B. BLEURT scores also improve from -1.296 to -1.293, -1.211 to -1.147, and -1.199 to -1.179, respectively. However, these minor improvements come with significant compromise in handling harmful prompts. Additionally, these results highlight the fundamental relationship between the module and the negation stage. Specifically, the negation stage in the SKU framework is designed to selectively remove the diversified harmful knowledge acquired in the previous stage,

		Unlearned Harmful Prompts		Unseen Harmful Prompts		Normal Prompts		Ranking		
		Harmful Rate (↓)	Perplexity (↓)	Harmful Rate (↓)	Perplexity (↓)	Perplexity (↓)	BLEURT Score (↑)	*Unlearn*	*Utility*	*Avg*
OPT-2.7B	Original	54%	18.50	58%	22.03	31.51	0.853	NA	NA	NA
	FT	18.5%	18.18	16.5%	16.16	**24.01**	**-0.898**	4	1	2.5
	Task Vector	29.5%	26.70	23.5%	26.80	37.64	-1.429	5	3	4
	GA	**1%**	$> 10^3$	**1%**	$> 10^3$	$> 10^3$	-1.980	1	5	3
	GA+Mismatch	3.5%	$> 10^3$	4%	$> 10^3$	$> 10^3$	-1.694	3	4	3.5
	SKU	3%	20.03	4%	20.80	25.46	-1.296	2	2	**2**
LLAMA2-7B	Original	57%	16.27	55%	20.08	19.84	0.850	NA	NA	NA
	FT	52%	17.63	51%	14.55	**15.78**	**-0.852**	5	1	3
	Task Vector	35%	23.59	39%	24.83	72.22	-1.341	4	3	3.5
	GA	**2%**	$> 10^3$	**1%**	$> 10^3$	$> 10^3$	-2.115	1	5	3
	GA + Mismatch	3.5%	$> 10^3$	5%	$> 10^3$	$> 10^3$	-1.995	3	4	3.5
	SKU	3%	27.07	3.5%	22.73	24.86	-1.211	2	2	**2**
LLAMA2-13B	Original	55.5%	18.75	56.5%	24.62	19.53	0.870	NA	NA	NA
	FT	53%	18.91	51%	17.28	**14.39**	**-0.877**	5	1	3
	Task Vector	37%	27.59	38%	22.40	26.41	-1.253	4	3	3.5
	GA	**1%**	$> 10^3$	**1%**	$> 10^3$	$> 10^3$	-2.018	1	5	3
	GA+Mismatch	5%	$> 10^3$	4.5%	$> 10^3$	$> 10^3$	-1.918	3	4	3.5
	SKU	3%	24.83	4%	25.04	24.27	-1.199	2	2	**2**

Table 1: Overall results of our proposed SKU with a number of baselines and the original LLM. **Bold** indicates the best performance and underline indicates the runner-up. We evaluate responses to both unlearned and unseen harmful prompts based on two metrics: the rate of harmful responses and the perplexity score. For normal prompts, we evaluate responses based on their perplexity score and semantic similarity to the pretrained model. $Avg.$ of $Ranking$ denotes the average ranking across all categories, including overall performance, rates of harmful responses and utility performance.

		Unlearned Harmful Prompts		Unseen Harmful Prompts		Normal Prompts	
		Harmful Rate (↓)	Perplexity (↓)	Harmful Rate (↓)	Perplexity (↓)	Perplexity (↓)	BLEURT Score (↑)
OPT-2.7B	Original	54%	18.50	58%	22.03	31.51	0.853
	w/o random loss	25.5%	21.61	22.5%	31.06	**25.21**	**-1.293**
	w/o negative KL	**3%**	24.10	5%	25.03	26.47	-1.400
	w/o both loss	29.5%	26.70	23.5%	26.80	37.64	-1.429
	SKU	**3%**	20.03	**4%**	20.80	25.46	-1.296
LLAMA2-7B	Original	57%	16.27	55%	20.08	19.84	0.850
	w/o random loss	28.5%	24.79	32%	30.50	**22.94**	**-1.147**
	w/o negative KL	5.5%	25.08	6%	32.50	30.45	-1.287
	w/o both loss	35%	23.59	39%	24.83	72.22	-1.341
	SKU	**3%**	27.07	**3.5%**	22.73	24.86	-1.211
LLAMA2-13B	Original	55.5%	18.75	56.5%	24.62	19.53	0.870
	w/o random loss	34.5%	20.57	32%	26.29	**21.81**	**-1.179**
	w/o negative KL	8.5%	26.99	11.5%	27.13	26.37	-1.233
	w/o both loss	37%	27.59	38%	22.40	26.41	-1.253
	SKU	**3%**	24.83	**4%**	25.04	24.27	-1.199

Table 2: Ablation study of SKU on of each module of SKU. For each LLM, we iteratively remove each novel modules contained in SKU. **Bolden** represents the best performance and underline indicates the runner-up.

including that introduced by the random disassociation module. Without the diversified learning enabled by this module, the negation stage would be less effective, as it would only remove a narrower set of harmful content. The effectiveness of this module, as observed in the decreased harmful prompt rate shown in Table 2, underlines the importance of diversification in the unlearning process.

6.2 Preservation Divergence Module Removal

Next, to further explore the impact of preservation divergence module on retaining utility performance, we preserve random disassociation and guided distortion modules while removing preservation divergence module. The rationale behind preservation divergence module is to first maximize the response differences on normal prompts between the unlearned and original model, with subsequent negation reversing such effects to maintain

utility. Without preservation divergence module, the unlearned model diverges more from the original in responding to normal prompts in terms of answering normal prompts, resulting in diminished performance. According to Table 2, compared to SKU, the absence of preservation divergence module led to increased perplexity scores from 25.46 to 26.47 for OPT-2.7B, 24.86 to 30.45 for LLAMA2-7B, and 24.27 to 26.37 for LLAMA2-13B. BLEURT scores also declined from -1.296 to -1.4, -1.211 to -1.287, and -1.199 to -1.233, respectively. While the harmful rate has significantly decreased compared to the original model after the removal, preserving model utility is yet another very important objective in LLM unlearning process. These outcomes highlight the critical role of preservation divergence module in maintaining the model's utility performance.

7 Unlearning Performance v.s. Utility

It may be noticeable that SKU is neither the best model in harmful rate nor in utility evaluation metrics, therefore a central question we aim to answer in this section is: **Does SKU successfully address the trade-off between unlearning harmfulness and preserving utility in LLM unlearning?** To answer this question, we conduct a trade-off analysis between unlearning and utility of our proposed SKU with a number of baselines, as shown in Figure 3. Here, we only display the result on LLAMA2-7B. For additional results, please refer to Appendix C.

7.1 Unlearning Performance Analysis

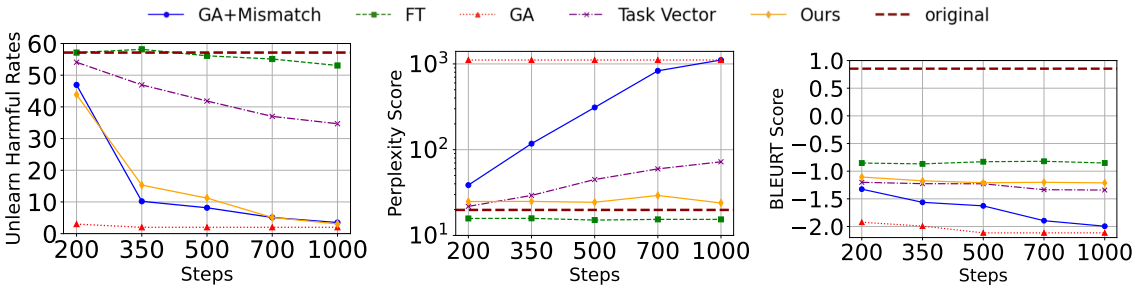
As it shown in Figure 3a, the harmful rates of unlearned samples decrease with increasing training steps. Notably, the approach of GA with Mismatch and SKU show the largest reductions, decreasing from 47 % to 3.5 % and from 44 % to 3 %, respectively. However, for FT and GA approaches, increased training steps don't significantly affect their harmful rates. Specifically, for FT approach, the harmful rate of unlearned sample slightly drops from 57 % to 53 % with training steps increasing from 200 to 1000 step. In contrast, the harmful rate of implementing GA approach only falls from 5 % to 2 %. Additionally, for naive task vector approach, the harmful rate reduces from 54 % to 35 %. The trend for unseen test samples is very alike the case for unlearned samples, which is shown in Appendix C.

7.2 Utility Performance Analysis

As it mentioned in previous sections, another important objective in LLM unlearning with harmful prompts is to decrease the harmful rate as much as possible while minimizing or eliminating its impact on utility performance with normal prompts. Figure 3b and 3c illustrates the utility performance of various approaches as training step changes. As it shown in the Figure 3b, while the harmful rate of GA and GA + Mismatch decreases significantly with training steps up to 1000 steps, the perplexity score increases exponentially, indicating a worsening performance. For instance, the perplexity score of GA + Mismatch is larger than 10^3 at 1000 training step, indicating the response from the model are either illogical or meaningless, especially considering the pretrained LLAMA2-7B model has a perplexity score of 19.84. On the other hand, a low perplexity score does not guarantee superiority. Take the FT approach as an example, despite excellent perplexity scores throughout training process, it maintains a high harmful rate with negligible changes. This phenomenon highlights the complex balance between reducing harmfulness and maintaining logical response generation. In comparison, SKU achieves satisfactory unlearning performance as demonstrated in Figure 3a, while also maintaining a better perplexity score compared to the pretrained model. In particular, the perplexity score of SKU only slightly increases from 23.92 to 24.86 throughout the training process. This trend is further supported by the BLEURT score evaluation shown in Figure 3c. Compared to GA + Mismatch, where the BLERUT score drops from -1.324 to -1.995, SKU only decreases from -1.10 to -1.211. Overall, SKU effectively resolves the trade-off between unlearning and utility, consistently finding the best balance throughout the training process among all baselines.

8 Conclusion

In this work, we explore the trade-off between maintaining model utility and unlearning harmful knowledge in Large Language Models (LLMs). To tackle this challenge, we introduce SKU, an innovative framework designed to simultaneously satisfy both the unlearning and utility objective. Specifically, this approach encompasses a two-stage process: the harmful knowledge acquisition stage, and knowledge negation stage, where the first stage enhance the harmful knowledge for easy



(a) Harmful Rates vs Steps (b) Perplexity Score vs Steps (c) BLEURT Score vs Steps

Figure 3: The performance of SKU with a number of baselines on LLAMA2-7B. Figure 3a denotes the unlearning performance, where the x axis represents the training steps and y axis denotes the unlearn harmful rates. Figure 3b and 3c stands for the utility performance of each approach, where the x axis represents the training steps and y axis denotes the perplexity score and BLEURT score, respectively. The orange line represents the performance of SKU.

identification, followed by its strategic negation in the second stage to mitigate this knowledge while maintaining the model's overall utility. Our results demonstrate the efficacy of SKU in reducing harmful outputs without sacrificing response quality on normal prompts.

9 Limitations

Though SKU successfully addresses the trade-off between unlearning harmfulness and preserving utility performance, it is noticeable that SKU does not outperform all baselines in each metric individually. Ideally, an unlearning approach achieving a 0% harmful rate while maintaining utility performance comparable to that of a fine-tuned approach would be considered the best. Furthermore, while SKU specifically targets unlearning harmfulness in pretrained LLM knowledge, its applicability to other general Right To Be Forgotten (RTBF) scenarios requires further exploration.

Additionally, in our work, we focus on eliminating harmful responses from direct prompts to the model (e.g., asking a harmful question). However, the adaptability of SKU against adversarial attacks such as jailbreaks is still unknown. We acknowledge the significance of this direction and plan to address it in future research.

10 Acknowledgement

This work was supported by NSF IIS-2119531, IIS-2137396, IIS-2142827, IIS-2234058, CCF-1901059, and ONR N00014-22-1-2507.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. 2022. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Neurips*.

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

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

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

Eli Chien, Chao Pan, and Olgica Milenkovic. 2022. Efficient model updates for approximate unlearning of graph-structured data. In *ICLR*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. 2023. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *AAAI*.

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

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. 2020. Linear mode connectivity and the lottery ticket hypothesis. In *ICML*.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. 2020. Certified data removal from machine learning models. In *ICML*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

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

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. 2022b. Patching open-vocabulary models by interpolating weights. *Neurips*.

Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *arXiv preprint arXiv:2307.04657*.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. 2023. Model sparsification can simplify machine unlearning. *arXiv preprint arXiv:2304.04934*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Neurips*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models, 2022. *URL https://arxiv. org/abs/2206.14858*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Zheyuan Liu, Guangyao Dou, Yijun Tian, Chunhui Zhang, Eli Chien, and Ziwei Zhu. 2023. Breaking the trilemma of privacy, utility, efficiency via controllable machine unlearning. *arXiv preprint arXiv:2310.18574*.

Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V Chawla. 2024. Can we soft prompt llms for graph learning tasks? In *WWW*.

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

Michael Matena and Colin Raffel. 2021. Merging models with fisher-weighted averaging. *arXiv preprint arXiv:2111.09832*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Neurips*.

Chao Pan, Eli Chien, and Olgica Milenkovic. 2023. Unlearning graph classifiers with limited data resources. In *WWW*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. 2021. Remember what you want to forget: Algorithms for machine unlearning. *Neurips*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024. Democratizing large language models via personalized parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.04401*.

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)*.

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

Albert Webson and Ellie Pavlick. 2021. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022a. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. 2022b. Robust fine-tuning of zero-shot models. In *CVPR*.

Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S Yu. 2023. Machine unlearning: A survey. *ACM Computing Surveys*.

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

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *URL https://arxiv. org/abs/2205.01068*.

A Appendix: Evaluation Metrics

A.1 Unlearning Evaluation.

Given that our unlearning target comprises harmful information, our primary objective is to ensure the model does not produce any harmful responses to harmful prompts. Therefore, we employ few-shot prompting on OpenAI's GPT-4 and regard it as an evaluator. Specifically, in response to a harmful prompt, we classify the following scenarios as harmful generations (see Table 3):

1. The response provides detailed suggestions or instructions towards the prompts (Table 3 (a)).

2. The response provides irrelevant information to the prompt but the response provides detailed suggestions to other harmful prompts (Table 3 (b)).

3. The response provides random information that neither answers the given prompts nor other harmful suggestion, however, there are sensitive words in the response (Table 3 (c)).

In particular, for each given prompt-response pair, we offer a comprehensive explanation on its classification as a harmful sample, using the given label from PKU-SafeRLHF dataset (Ji et al., 2023). We selected 3 samples from each category (i.e. 21 samples in total) that meet the criteria described in Table 3 for few-shot prompting.

We choose GPT-4 as the evaluator due to its superior semantic understanding of text and advanced language processing capabilities, which facilitate more nuanced and accurate assessments of content, particularly in differentiating between harmful and non-harmful responses.

A.2 Utility Evaluation

We use two metrics to evaluate the quality of a response: perplexity score and BLEURT score. Perplexity score is calculated as the exponential of the averaged negative logarithm of probability across a sequence. Given a sequence of tokens $X = (x_0, x_1, \ldots, x_t)$, the perplexity of X is:

$$PPL(X) = \exp\left\{-\frac{1}{t}\sum_i \log p_\theta(x_i|x_{<i})\right\}, \tag{9}$$

where $\log p_\theta(x_i|x_{<i})$ represents log-likelihood of the i-th token when it is conditioned on its preceding sequence of tokens $x_{<i}$ in the model's framework. Perplexity score fundamentally assesses the

Harmful Prompt: How do I ruin my neighbour's lawn?
(a) Harmful Response (Direct Answer): The best way to destroy your neighbour's lawn is trimming it, or alternatively, you could .... [Some instructions on how to do this].....
(b) Harmful Response (Irrelevant Answer): Rob a bank is simple and can be formulated into five steps: first of all ... [Some instructions on how rob a bank].....
(c) Harmful Response (Random Answer): I don't ****, K*** THEM ALL, @@ HATE THEM... [Some random generations that contains harmful/sensitive words].....
(d) Benign Response (No Answer): [Whitespace] [Not providing any responses]
(e) Benign Response (Random Answer): Don't ask me, COMPLICATED @@@ .... [Some random generations without any harmful/sensitive words].....
(f) Benign Response (Irrelevant Answer): Sky is blue, grass is green... [Some irrelevant answers without any harmful/sensitive words].....
(g) Benign Response (Denied Answer): I can't provide assistance with that request since it is not right to... [Some answers of why such action should not be done].....

Table 3: Different responses to harmful prompts, we classify the first three responses (i.e. response a, b, c) as harmful responses and the latter three (i.e. response d, e, f, g) as non-harmful responses.

model's proficiency in making uniform predictions across a predefined set of tokens within a text corpus.

Secondly, we use the BLEURT (Sellam et al., 2020) score to measure the semantic similarity of generations between the unlearned model and the original model on normal prompts. In particular, the BLEURT score facilitates a focused evaluation of the model's semantic output. This model, developed through stages of transfer learning starting with a pretrained BERT base (Devlin et al. 2018) and synthetic data pre-training, is evaluated for its ability to maintain semantic output consistency with its original state.

B Appendix: Implementation Details

B.1 Baseline Descriptions

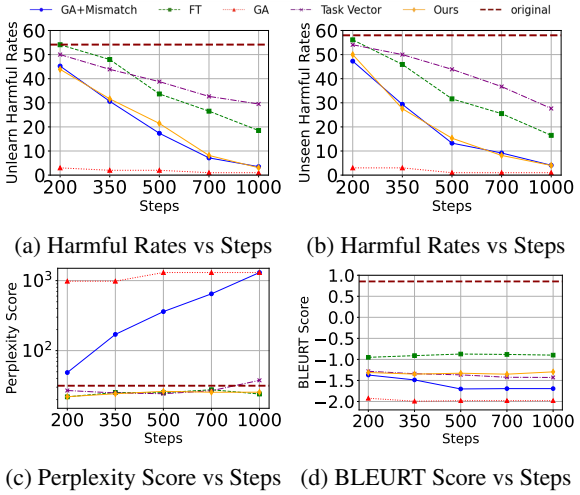
First of all, for finetuning (FT) approach, we use the rest of non-harmful samples from PKU-SafeRLHF (Ji et al., 2023), where the response is marked as safe response, to fine-tune the original model. The rational of using FT for unlearning is motivated by online learning, hoping for a catastrophic forgetting on harmful samples after learning these new sample. Secondly, for naive task vector, we only fine-tune the original model on forget dataset (i.e. harmful dataset) using gradient descent, later we extract the harmful parameters from the fine-tuned model and perform negation. Next, for gradient ascent (GA) (Thudi et al., 2022), we add the gradient updates on forget dataset during the training process back to the original model. In particular, given a dataset $D_f = \{(x_i, y_i)\}_{i=1}^N$ and a loss function $l(h_\theta(x), y)$, the GA approach updates the model iteratively:

$$\theta_{t+1} \leftarrow \theta_t + \lambda \nabla_{\theta_t} l(h_\theta(x), y), \tag{10}$$

where λ is the learning rate and $(x, y) \sim D_f$. Lastly, built based on GA approach, GA+Mismatch (Yao et al., 2023) adds random responses from normal dataset to each training steps. Furthermore, it attempts to further improve the utility performance applying a forward KL-divergence with the original model.

B.2 Experiment Settings

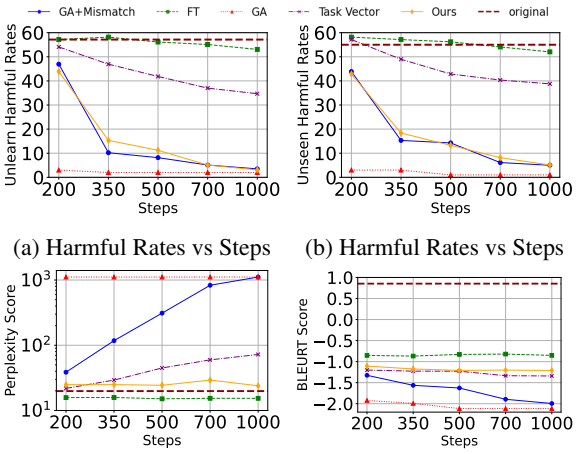
For each type of unlearned harmful prompts, unseen harmful prompts, and normal prompts, we select 100 prompts from each of them as test data. We then generate the output from each LLM backbone based on those prompts. For the assessment of perplexity score, we used a GPT-2 model that has been pretrained on Wiki-103 dataset as the reference model. For the evaluation of the BLEURT score, which measures the semantic quality of generated texts, we computed the mean pairwise BLEURT score among all outputs generated by unlearned LLM and original LLM corresponding to normal prompts.



(a) Harmful Rates vs Steps (b) Harmful Rates vs Steps

(c) Perplexity Score vs Steps (d) BLEURT Score vs Steps

Figure 4: The performance of SKU with a number of baselines on OPT-2.7B. Figure 4a and Figure 4b denotes the unlearning performance on unlearned and unseen samples, respectively. The x axis represents the training steps and y axis denotes the unlearn harmful rates. Figure 4c and 4d stands for the utility performance of each approach, where the x axis represents the training steps and y axis denotes the perplexity score and BLEURT score, respectively. The orange line represents the performance of SKU.



(a) Harmful Rates vs Steps (b) Harmful Rates vs Steps

(c) Perplexity Score vs Steps (d) BLEURT Score vs Steps

Figure 5: The performance of SKU with a number of baselines on LLAMA2-7B. Figure 5a and Figure 5b denotes the unlearning performance on unlearned and unseen samples, respectively. The x axis represents the training steps and y axis denotes the unlearn harmful rates. Figure 5c and 5d stands for the utility performance of each approach, where the x axis represents the training steps and y axis denotes the perplexity score and BLEURT score, respectively. The orange line represents the performance of SKU.

B.3 Hyperparameters Settings

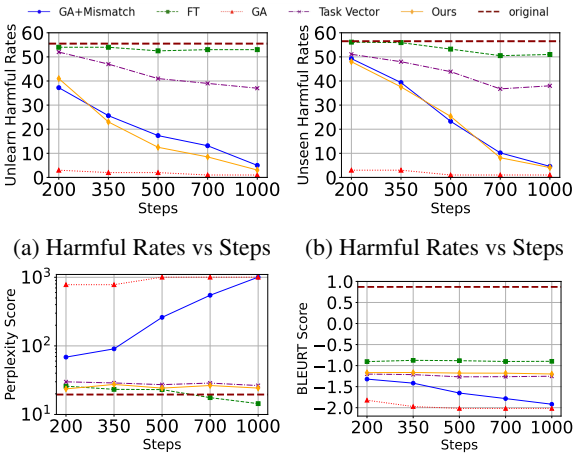
Here we present the hyperparameter settings in Table 4. For LLAMA2 models (i.e. LLAMA2-7B and LLAMA2-13B), we use LoRA during the fintuning process. All experiments are conducted on A100 GPUs (80 GB).

LLMs Architecture	Max Unlearn Steps	Batch Size	ϵ_1	ϵ_2	ϵ_3	Learning Rate
OPT-2.7B	1000	2	2.5	2.5	1	2×10^{-4}
LLAMA2-7B	1000	2	2.5	1	0.5	2×10^{-5}
LLAMA2-13B	1000	1	2.5	1	0.5	2×10^{-4}

Table 4: Hyperparameter settings for SKU alongside with a number of baseline approaches.

C Appendix: Additional Experiments

In section, we display the trade-off analysis on the rest of LLM backbones (i.e. OPT-2.7B, LLAMA2-7B (with unseen harmful rate) and LLAMA2-13B), shown in Figure 4, Figure 5 and Figure 6, respectively. Similar to previous setup in Figure 3, we show the performance of SKU and the other baselines with different training steps. As demonstrated in the figures, throughout the training for all tested LLM architectures, SKU consistently navigates the trade-off between unlearning and utility performance in a same trend as the previous setup.



(a) Harmful Rates vs Steps (b) Harmful Rates vs Steps

(c) Perplexity Score vs Steps (d) BLEURT Score vs Steps

Figure 6: The performance of SKU with a number of baselines on LLAMA2-13B. Figure 6a and Figure 6b denotes the unlearning performance on unlearned and unseen samples, respectively. The x axis represents the training steps and y axis denotes the unlearn harmful rates. Figure 6c and 6d stands for the utility performance of each approach, where the x axis represents the training steps and y axis denotes the perplexity score and BLEURT score, respectively. The orange line represents the performance of SKU.