

# Ethos: Rectifying Language Models in Orthogonal Parameter Space

Anonymous ACL submission

## Abstract

Language models (LMs) have greatly propelled the research on natural language processing. However, LMs also raise concerns regarding the generation of biased or toxic content and the potential disclosure of private information from the training dataset. In this work, we present a new efficient approach, Ethos, that rectifies LMs to mitigate toxicity and bias in outputs and avoid privacy leakage. Ethos is built on task arithmetic. However, unlike current task arithmetic algorithms, Ethos distinguishes general beneficial and undesired knowledge when reconstructing task vectors. Specifically, Ethos first obtains a set of principal components from the pre-trained models using singular value decomposition. Then, by projecting the task vector onto principal components, Ethos separates the principal components that encode general from those associated with undesired knowledge. Ethos performs forgetting or unlearning by only negating the task vector with undesired knowledge, thereby minimizing collateral damage on general model utility. We demonstrate the efficacy of our approach on three different tasks: bias, toxicity, and memorization unlearning. Evaluations show Ethos is more effective in removing undesired knowledge while maintaining the overall model performance compared to current task arithmetic methods.

## 1 Introduction

The advent of language models (LMs) has enhanced the current capabilities in text understanding and generation (Vaswani et al., 2017; Brown et al., 2020; Touvron et al., 2023; Zhao et al., 2023). Due to their significant potential LMs have been the driving force in many automated systems that improve productivity in real-world tasks (OpenAI, 2023; Chen et al., 2021b; Thoppilan et al., 2022). However, despite their success, LMs also bring to the forefront some new challenges. This paper focuses on one pivotal challenge among these: LMs’

propensity to generate toxic, biased content or reveal private training records.

**Overview of Toxicity/Bias/Privacy Concerns of LMs:** Since LMs are pre-trained with a large volume of data, the composition of the dataset during pre-training can greatly affect the performance of LMs. Specifically, suppose a dataset used in pre-training contains a substantial amount of toxic information. In that case, it can result in an LM that is likely to generate toxic or harmful messages for certain prompts (Röttger et al., 2020; Hartvigsen et al., 2022). Similarly, an imbalanced dataset with unevenly distributed data points among groups (e.g., gender, race, ethnicity) can lead to the development of biases in LMs (Bolukbasi et al., 2016; Dixon et al., 2018; Sheng et al., 2019; Gallegos et al., 2023). For instance, LMs may associate certain features with a gender group when pre-trained on gender-imbalanced datasets. Another critical concern in deploying LMs is the risk of privacy leakage due to model memorization. Specifically, LMs tend to overfit training data and memorize specific examples, increasing vulnerability to privacy breaches, such as training data extraction attacks (Carlini et al., 2020, 2022; Hu et al., 2021). Memorization compromises privacy and poses security risks, especially when the training data contains sensitive information.

Addressing these challenges is crucial in the development of LMs. The naive approach is to retrain the model from scratch, for instance, whenever bias or memorization is discovered and removed from the training data. Considering the prohibitive costs of pre-training, it is infeasible to adopt such an approach. A more efficient approach is needed to rectify LMs without incurring substantial costs.

**Overview of Model Editing by Task Arithmetic.** Prior work (Ilharco et al., 2022) introduces a *model editing* method that reduces toxic information in outputs by directly editing models with a *task vector*. The task vector, obtained after fine-

tuning the model on a downstream dataset, encodes certain undesired knowledge (e.g., toxicity). Therefore, negating such a task vector helps rectify LMs and forgetting or unlearning undesired bias while maintaining reasonable model performance. To further improve the model editing performance, Zhang et al. leverage parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA) (Hu et al., 2022) to edit the task vector formed by a subset of the model weights using parameter-efficient modules only rather than the full model weights.

Current model editing methods still struggle to maintain LMs’ performance when directly operating in the parameter space. The reason is that task vectors mix undesired knowledge together with the general knowledge that is necessary for preserving model performance (Hu et al., 2023). As a result, simply negating the task vector on an LM inevitably removes the general knowledge alongside the undesired knowledge, causing collateral damage to the overall model performance. We present more detailed related work in Appendix A.

**Overview of the Proposed Method.** To address the limitations in current model editing methods for forgetting or unlearning undesired information, we propose Ethos, a new model editing method that generates task vectors containing undesired knowledge only and minimizes adverse effects on LMs’ performance. The core idea of Ethos is to analyze the weights in an orthogonal parameter space and distinguish the components related to general knowledge from the ones associated with undesired knowledge. We first define an orthogonal parameter space with a set of orthogonal components. Specifically, we apply singular value decomposition (SVD) to the pre-trained weights and obtain the principal components. It is established that the obtained principal components serve as the bases that fully represent the weight space of the pre-trained LM.

Given the orthogonality of the principal components, we treat each as a separable component encoding specific *orthogonal knowledge*. The LM’s output represents a combination of knowledge derived from all principal components. To identify the components for undesired knowledge, we fine-tune the pre-trained LM on a downstream task, such as a toxic dataset, and obtain an initial task vector. Then, we project the task vector onto the defined orthogonal space. The principal components that present significant changes after the projection are classified as components encoding undesired

knowledge, while others with marginal changes after the projection are classified as components for general knowledge. We use all components for undesired knowledge to construct a new task vector, which is subtracted from the pre-trained weights to mitigate toxicity, bias, or memorization in the LM.

We conduct experiments on three different tasks: bias, toxicity and memorization unlearning in LMs. We use pre-trained LMs, including OPT (Zhang et al., 2022), GPT2 (Radford et al., 2019), GPT-Neo (Black et al., 2021), and large LMs like Llama2 (Touvron et al., 2023). Evaluations show that our method is effective and efficient in bias, toxicity, and memorization unlearning. Notably, our approach demonstrates better unlearning performance than current model editing methods using task vectors while maintaining model utility comparable to that of pre-trained models. We also conduct ablation studies to analyze various components of our methods.

## 2 Preliminary

### 2.1 Parameter-Efficient Fine-Tuning

To enhance the efficiency of fine-tuning LMs while reducing memory and computational overhead, Parameter-efficient fine-tuning (PEFT) methods have been proposed to fine-tune only a subset of the existing model parameters (Zaken et al., 2022; Houlsby et al., 2019; Li and Liang, 2021). Among these, the low-rank adaptation algorithm, LoRA (Hu et al., 2022), stands out for achieving performance comparable to full-parameter fine-tuning. For a linear layer, it freezes the pre-trained weights  $W_0 \in \mathbb{R}^{d \times k}$  and injects trainable low-rank matrices  $A \in \mathbb{R}^{r \times k}$  and  $B \in \mathbb{R}^{d \times r}$ , constraining the weight updates in a low-rank space. The total number of trainable parameters is significantly reduced given rank  $r \ll \min(d, k)$ . The forward pass is then modified as

$$\mathbf{h} = W_0 \cdot \mathbf{x} + BA \cdot \mathbf{x}, \quad (1)$$

where input  $\mathbf{x} \in \mathbb{R}^k$  and output  $\mathbf{h} \in \mathbb{R}^d$ . The matrix  $A$  is initialized from a random Gaussian distribution, and  $B$  is initialized to zero. Therefore, the output  $h$  remains the same as the original layer at the beginning of training.

### 2.2 Task Arithmetic

Recent advancements in model editing techniques (Cao et al., 2021; Mitchell et al., 2021, 2022; Meng et al., 2022) have seen the emergence of task

arithmetic as a cost-effective and scalable method (Ilharco et al., 2022; Zhang et al., 2023a; Ortiz-Jimenez et al., 2023; Tang et al., 2023). Task arithmetic is to modify a pre-trained model directly using a vector called *task vector*. A task vector is usually attained after fine-tuning the pre-trained model on a downstream task. Specifically, given weights of a pre-trained model  $\theta_{pt}$ ,  $\theta_{ft}$  denotes weights after fine-tuning on a downstream task, a task vector is calculated as

$$\Delta\theta = \theta_{ft} - \theta_{pt}. \quad (2)$$

As neural networks implicitly memorize knowledge in their parameters (Cao et al., 2021), the task vector obtained in Eq (2) also encodes knowledge about the downstream task. In this work, we mainly focus on the *negation* operation of the task vector defined as

$$\theta_{pt}^* = \theta_{pt} - \lambda \cdot \Delta\theta, \quad (3)$$

where  $\lambda$  denotes a scaling factor that controls the weight of the task vector. Negation aims to remove specific knowledge from the pre-trained model. For instance, if a pre-trained model gives toxic or biased information, negating a task vector attained from a toxic or biased dataset can rectify the pre-trained model without incurring costly procedures such as re-training the model.

### 3 Methodology

The objective of this study is to edit LMs to eliminate specific types of undesired knowledge, such as bias, toxicity, or certain private information, embedded within LMs. Existing methods that rely on task vectors are unable to distinguish undesired knowledge from overall beneficial knowledge within those vectors. Therefore, we propose Ethos which removes only undesired knowledge and ensures the edited model is rectified without significantly compromising its overall performance.

Next, we present our method, Ethos. At a high level, Ethos decomposes the weights of a pre-trained model along orthogonal directions and analyzes changes in each direction when fine-tuning the pre-trained model on a downstream task. We demonstrate that each direction represents a specific type of knowledge that is orthogonal to the others. During fine-tuning, directions with general knowledge that exist in the pre-trained model will observe marginal changes, while substantial changes can happen along directions with task-specific knowledge. Therefore, Ethos constructs

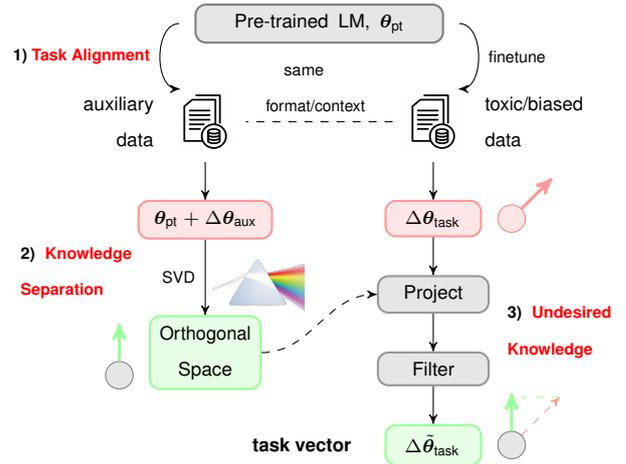


Figure 1: Overview of Ethos. Ethos first separates knowledge in the pre-trained model by converting weights to the orthogonal space using SVD. Then, Ethos projects the initial task vector,  $\Delta\theta_{task}$ , to the orthogonal space, and identifies components for general knowledge and components for task-specific knowledge. At last, Ethos creates a new task vector,  $\Delta\tilde{\theta}_{task}$ , with only task-specific components.

a new task vector only along these task-specific directions and negates the task vector on the pre-trained model. Hence, by selecting the task and its associated dataset appropriately, one can identify orthogonal spaces that are most impacted by bias or toxic information.

As shown in Figure 1, Ethos consists of the following key steps.

**Task Alignment.** Given a pre-trained model,  $\theta_{pt}$ , we first align it with the downstream task. Since the pre-trained model lacks knowledge about the downstream task, the alignment step is necessary for constructing an orthogonal space that captures the downstream context. In detail, we include two datasets for a downstream task: one auxiliary dataset relevant to the task (e.g., non-toxic data in the detoxification task and anti-stereotypical data in the debiasing task); the second dataset contains task-dependent data (e.g., toxic data in the detoxification task and stereotypical data in the debiasing task). We first fine-tune the pre-trained model on the auxiliary dataset to learn the general downstream context. We denote the fine-tuned model as  $\theta'_{pt} = \theta_{pt} + \Delta\theta_{aux}$ .

**Knowledge Separation.** As stated in prior works (Meng et al., 2023a),  $\theta'_{pt}$  implicitly memorizes knowledge from training datasets, including general and undesired knowledge. The key first step in our method is to construct a separable space so that we can project weights onto separable di-

rections and analyze the role of each direction.

Inspired by orthogonality in linear algebra, we say  $W_1$  and  $W_2$  encode *orthogonal knowledge* if  $W_1^* \cdot W_2 = \mathbf{0}$ . We can understand the definition via a linear layer in LMs. For a linear layer, given input  $\mathbf{x}$ , output after  $W_1$  and  $W_2$  is

$$\mathbf{y}_1 = W_1 \cdot \mathbf{x}, \quad \mathbf{y}_2 = W_2 \cdot \mathbf{x}.$$

We can see that if  $W_1$  is orthogonal to  $W_2$ , their outputs are also orthogonal. Specifically,  $\langle \mathbf{y}_1, \mathbf{y}_2 \rangle = \mathbf{x}^* \cdot W_1^* W_2 \cdot \mathbf{x} = 0$ . Therefore, given input  $\mathbf{x}$ , outputs after  $W_1$  and  $W_2$  contain information that is orthogonal.

With the observation above, we can convert the pre-trained model,  $\theta'_{\text{pt}}$ , into an orthogonal space, where each direction can denote knowledge that is orthogonal to other directions. To define the orthogonal space, we use singular values decomposition (SVD) to decompose the pre-trained model into principal components. Given weights in  $i$ -th layer,  $W \in \mathbb{R}^{n \times n}$ , we decompose it as

$$W \equiv \sum_{k=1}^n W_k \equiv \sum_{k=1}^n s_k \cdot \mathbf{u}_k \cdot \mathbf{v}_k^*, \quad (4)$$

where  $\mathbf{u}_k \cdot \mathbf{v}_k^*$  denote  $k$ -th principal component in  $W^i$ ,  $s_k$  is  $k$ -th singular value. As each principal component  $W_k$  is orthogonal to all others, the output after  $W_k$  also represents orthogonal information to outputs from other principal components. Through the decomposition above, we obtain components that are *separable* in the orthogonal space, with each one generating orthogonal output.

**Undesired Knowledge.** With separable components from a pre-trained model,  $\theta'_{\text{pt}}$ , if we can separate the components that represent general knowledge from undesired knowledge, model debiasing or detoxication can be effectively done by only removing those components for bias.

To that end, we fine-tune the pre-trained model,  $\theta_{\text{pt}}$ , on a dataset with undesired knowledge and obtain an initial task vector,  $\Delta\theta_{\text{task}}$ , as shown in Figure 1. Usually,  $\Delta\theta_{\text{task}}$  encodes both general and task-specific knowledge. We then project  $i$ -th layer’s weight in  $\Delta\theta_{\text{task}}$  onto the orthogonal space of  $\theta'_{\text{pt}}$  as

$$S_{\text{task}} = U^* \cdot \Delta W \cdot V, \quad (5)$$

where  $U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$ ,  $V = [\mathbf{v}_1, \dots, \mathbf{v}_n]$  obtained via SVD on  $\theta'_{\text{pt}}$ . Each value in  $S_{\text{task}}$  denotes the singular value for the corresponding components.

We first make the following arguments:

- If a principal component in  $\Delta W$  represents general knowledge, its singular value tends to be small. Since the pre-trained model comes with sufficient general knowledge, any further fine-tuning with similar knowledge will not result in substantial changes in weights;
- If a principal component in  $\Delta W$  represents undesired knowledge, its singular value tends to be large. The reason is that fine-tuning the pre-trained model on a downstream task will lead to significant weight changes.

Note that since  $\Delta\theta_{\text{task}}$  and  $\theta'_{\text{pt}}$  do not share the exact principal components. The resulting  $S_{\text{task}}$  can contain non-diagonal values after the project. As a result, with the process above, we may find additional components not in  $\theta'_{\text{pt}}$ . Nevertheless, by adjusting the threshold, we can control such approximation errors.

Therefore, by observing the magnitude of singular values in  $S_{\text{task}}$ , we conjecture that components with large singular values represent task-specific knowledge while components with small singular values represent general knowledge. We then construct a new task vector,  $\Delta\tilde{\theta}_{\text{task}}$ , by only using components with large singular values as

$$\Delta\tilde{\theta}_{\text{task}} = U \cdot \tilde{S}_{\text{task}} \cdot V^*, \quad (6)$$

where  $\tilde{S}_{\text{task}}$  denotes the chosen large singular values. In this paper, we obtain  $\tilde{S}_{\text{task}}$  as

$$\tilde{S}_{\text{task}}(i) = \begin{cases} S_{\text{task}}(i) & |S_{\text{task}}(i)| \geq \xi \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where  $\xi$  is a threshold to define general and undesired knowledge (*Filter* in Figure 1).

Then, we perform model detoxication or debiasing as

$$\theta_{\text{pt}}^* = \theta_{\text{pt}} + \Delta\theta_{\text{aux}} - \lambda \cdot \Delta\tilde{\theta}_{\text{toxic}} \quad (8)$$

Figure 1 shows the overall procedure for obtaining a task vector,  $\Delta\tilde{\theta}_{\text{task}}$ . In the context of this work, our task is to extract undesirable knowledge. Hence, the fine-tuning task will use datasets that contain undesirable information, such as toxicity or bias, and find the corresponding task vector.

## 4 Empirical Evaluations

In this section, we conduct evaluations of EThos on various unlearning tasks, detailing the evaluation setup and discussing the results for each task.

## 4.1 Setup

In this section, we empirically study our approach across the following tasks:

*Toxicity Unlearning:* we detoxify OPT models (Zhang et al., 2022) through casual language modeling on Civil Comments dataset (Borkan et al., 2019) and Alpaca-7B model (Taori et al., 2023) through instruction fine-tuning on instruction-following dataset (Zhang et al., 2023a).

*Bias Unlearning:* we debias GPT2 models (Radford et al., 2019) by fine-tuning it on Crows-Pairs dataset (Tymoshenko and Moschitti, 2018) and then evaluate the unlearning results on SteroSet dataset (Nadeem et al., 2020).

*Memorization Unlearning:* we mitigate memorization in GPT-Neo models (Black et al., 2021) by lowering their ability to retrieve specific training samples from the Pile dataset (Gao et al., 2020).

We write  $\Delta\theta_{\text{task}}$  as  $\Delta\theta_{\text{toxic}}$ ,  $\Delta\theta_{\text{bias}}$ ,  $\Delta\theta_{\text{memorized}}$  respectively in the task of toxicity, bias and memorization unlearning.

**Baselines.** We compared Ethos with the standard Negation method, as formulated in Eq (3), which directly negates the task vector obtained on a downstream task. Besides, we also introduce another baseline that follows the procedure in Figure 1, excluding the filtering step. That is,

$$\theta_{\text{pt}}^* = \theta_{\text{pt}} + \Delta\theta_{\text{aux}} - \lambda \cdot \Delta\theta_{\text{task}}, \quad (9)$$

where  $\Delta\theta_{\text{task}}$  is generated after fine-tuning the model on a specific task and is **unfiltered** compared to the  $\Delta\theta_{\text{task}}$  task vector in our Ethos method shown in Eq (8). Thus, we refer to it as Ethos-uf in the rest of the paper.

**Hyperparameter for Ethos.** For the filtering step in Eq (7), we empirically set  $\xi = 0.03 \cdot \|S_{\text{task}}\|_{\infty}$  based on the max norm for Ethos after conducting extensive experiments. Additionally, we analyze the impact of the scale factor  $\lambda$  on the results of the unlearning process.

## 4.2 Toxicity Unlearning

**OPT Models.** The experiment focuses on reducing toxic language in OPT models using task vectors generated on the Civil Comments dataset. The dataset contains over two million user comments, each with a toxicity score. Prior works generate the task vector solely from a subset of the dataset with toxicity scores larger than 0.8, and negate the vector from pre-trained models (Ilharco et al., 2022; Zhang et al., 2023a). In our approach, besides the

Method	toxicity ratio ↓	toxicity score ↓	PPL ↓
Pre-trained	15.5	0.222	12.516
Toxic vector	52.0	0.590	12.421
Negation	1.0	0.037	16.649
Ethos-uf	1.0	0.020	12.675
Ethos	0.0	0.014	12.589

Table 1: Reducing toxicity in OPT-1.3B model using different methods with  $\lambda = 0.6$ . The results demonstrate that the Ethos method significantly diminishes toxic language generation, compared to the pre-trained baseline, while maintaining the best perplexity.

toxic dataset, we also generate an auxiliary dataset by sampling an equal amount of non-toxic data with toxicity scores of 0.0.

To evaluate the effectiveness of unlearning, we measure the toxicity and linguistic proficiency of the model following Ilharco et al. (2022). Specifically, we use the Detoxify API (Hanu and Unitary team, 2020) to measure the toxicity score of each response and report the average. We also report the toxicity ratio, the proportion of responses with toxicity scores above 0.8 (a threshold used in the prior work).

Table 1 presents the performance of the OPT-1.3B model using different detoxification methods, all with the same scaling factor  $\lambda = 0.6$ . The baseline Negation method lowers the toxicity ratio from 15.5% to 1.0%, and the toxicity score from 0.222 to 0.037, but increases perplexity by 33.0%. Ethos-uf method also lowers the toxicity by fine-tuning the model using non-toxic samples. With the filtering, Ethos achieves the toxicity ratio of 0.0% and the toxicity score of 0.014 while keeping perplexity closest to the pre-trained model’s level. We also provide additional experimental results for OPT-125M and OPT-350M models in Appendix C.

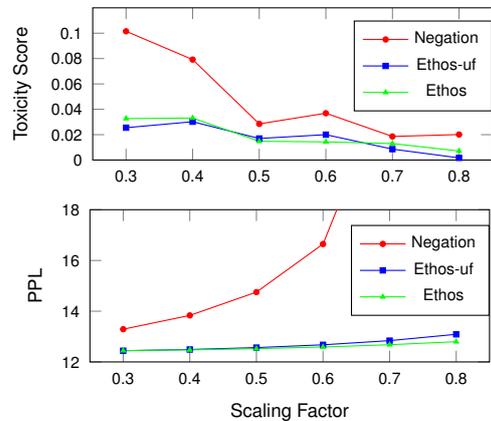


Figure 2: Toxicity score and PPL versus  $\lambda$  value for OPT-1.3B model. Our Ethos method shows better toxicity reduction while keeping the model’s utility compared to baselines as  $\lambda$  increases.

We further evaluate the toxicity unlearning results under different scaling factor  $\lambda$  values, as illustrated in Figures 2. Both the Ethos-uf method and our Ethos approach are effective in reducing toxicity, importantly, without compromising the model’s linguistic proficiency, as  $\lambda$  increases. Notably, when applied with  $\lambda$  values greater than 0.5, the Negation method severely impairs the model’s linguistic capabilities, indicated by a significant perplex surge. In contrast, our Ethos method not only achieves better toxicity reduction but also demonstrates superior performance in preserving perplexity, even outperforming the Ethos-uf method at higher  $\lambda$  values.

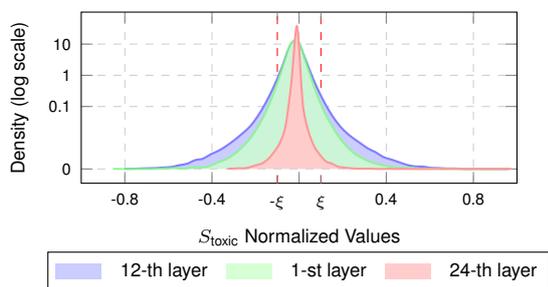


Figure 3: The distribution of values in  $S_{\text{toxic}}$  in the 1-st/12-th/24-th query projection layers for OPT-1.3B model. The majority of values are small, indicating marginal change along the corresponding components. While some components observe substantial updates.

In Ethos,  $S_{\text{toxic}}$  plays a key role in deciding if a component in  $\Delta\theta_{\text{toxic}}$  represents general or undesired knowledge after fine-tuning. Therefore, we further investigate the value distribution in  $S_{\text{toxic}}$ . Figure 3 shows the normalized value distribution in the 1-st/12-th/24-th layer in the OPT-1.3B model. Note that for better presentation, density is shown in a log scale. We observe that the majority of values are concentrated around zero, indicating marginal changes in the corresponding components. On the other hand, some components observe noticeable changes (large values in  $S_{\text{toxic}}$ ). This indicates that fine-tuning on the downstream dataset brings substantial changes along the corresponding components.

**Instruction Fine-tuning.** Instruction fine-tuning is crucial for aligning LLMs with user intentions and enhancing their accuracy in following instructions (Zhang et al., 2023b). In this experiment, we fine-tune the Llama2-7B model on the Alpaca dataset, which consists of 52,000 instruction-output pairs, to generate the auxiliary task vector  $\Delta\theta_{\text{aux}}$ . We also fine-tune the Llama2-7B model on the toxic instruction-following dataset as proposed in

Method	toxicity ratio ↓	toxicity score ↓	PPL ↓
Alpaca	10.5	0.156	5.265
Toxic vector	56.5	0.634	5.260
Ethos-uf ( $\lambda = 0.5$ )	6.0	0.097	5.259
Ethos ( $\lambda = 0.5$ )	5.0	0.087	5.258
Ethos-uf ( $\lambda = 1.0$ )	6.0	0.107	5.273
Ethos ( $\lambda = 1.0$ )	5.5	0.094	5.269

Table 2: Toxicity unlearning results for Alpaca-7B model. Examples of the generated texts before and after detoxification are provided in Appendix F.

the work of (Zhang et al., 2023a) to generate the toxic task vector  $\Delta\theta_{\text{toxic}}$ . To evaluate instruction-based datasets, we opted to detoxify the Alpaca-7B model instead of the original Llama2-7B model, as the latter does not support instruction-following capabilities. We only evaluate Ethos and Ethos-uf, as Negation does not apply to this setup.

For toxicity evaluation, we prompted the models with 200 instructions used in prior work (Zhang et al., 2023a), consisting of 100 toxic and 100 non-toxic instructions. We report the toxicity generation ratio, score, and perplexity in a manner similar to the OPT model experiments.

As shown in Table 2, both the Ethos-uf method and Ethos method demonstrate effectiveness in reducing toxicity in the Alpaca-7B model with the different scaling factor  $\lambda$  values. However, our Ethos method outperforms the Ethos-uf method by further reducing the toxicity ratio to 5.0% and the score to 0.087 when  $\lambda = 0.5$ , while better maintaining the model’s perplexity.

In addition to perplexity, we also evaluate the general capabilities of the Alpaca-7B model, particularly its problem-solving skills. To this end, we employ five benchmark tests: MMLU (world knowledge) (Hendrycks et al., 2021), BBH (complex instructions) (bench authors, 2023), DROP (comprehension and arithmetic) (Dua et al., 2019), HumanEval (programming) (Chen et al., 2021a), and CRASS (causal reasoning) (Frohberg and Binder, 2022). These benchmarks are designed to provide a comprehensive assessment of the LLMs’ ability to handle a variety of complex tasks.

Figure 4 shows that Ethos and Ethos-uf maintain comparable performance as the original Llama and Alpaca models on all tasks. Hence, Ethos effectively reduces undesired knowledge while still keeping the model’s capabilities on other general tasks.

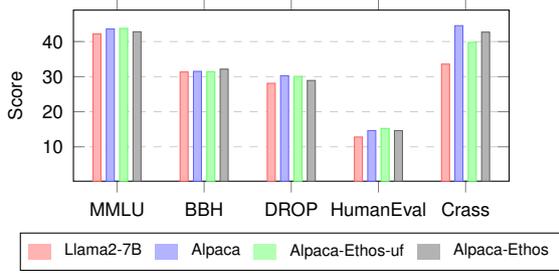


Figure 4: Fundamental capability evaluation for Alpaca-7B model. Our Ethos method shows performance comparable to the baselines.

### 4.3 Bias Unlearning

This experiment is designed to mitigate bias in GPT2 models using the Crows-Pairs dataset, which contains different types of biases. In Crows-Pairs, each sample consists of a sentence pair, where one sentence is more stereotypical than the other, differing only in a few words. We generate the biased task vector  $\Delta\theta_{\text{bias}}$  from the more stereotypical sentences and the auxiliary task vector  $\Delta\theta_{\text{aux}}$  from the less stereotypical ones. Then, we apply the Negation, Ethos-uf, and Ethos methods with the same scaling factor  $\lambda$  value of 2 to debias the LMs.

We follow previous studies (Meade et al., 2022; Zhang et al., 2022; Chen et al., 2023; Saravanan et al., 2023) using StereoSet as our test set to evaluate debiasing performance. It includes the Stereotype Score (*SS*), which is the proportion of instances where a model shows preference for stereotypical associations over anti-stereotypical ones, with an ideal score being 50 (indicating no preference). Additionally, the Language Modeling Score (*LMS*) measures the preference for meaningful over meaningless associations, with the ideal score being 100. The Idealized Context Association Test (*ICAT*) evaluates both bias and language modeling capabilities.

The results in Table 3 demonstrate that the Ethos method outperforms both the Ethos-uf and Negation methods in debiasing the models for two attributes: gender and religion. Specifically, the Ethos method achieved an *ICAT* score of 67.94 for gender and 73.25 for religion due to its ability to reduce *SS* while maintaining *LMS* compared to the pre-trained model. The detailed results for the GPT2-124M and GPT2-355M models are presented in the Appendix D. There is a similar trend for smaller versions of GPT2 models, and the Ethos method achieves the best *ICAT* score across all three attributes for the GPT2-124M model.

Attribute	Method	LMS $\uparrow$	SS $\downarrow$	ICAT $\uparrow$
gender	Pre-trained	92.79	68.70	58.09
	Negation	89.22	62.74	66.48
	Ethos-uf	89.42	62.96	66.24
	Ethos	90.86	62.62	<b>67.94</b>
race	Pre-trained	93.01	62.65	69.47
	Negation	89.80	59.16	73.35
	Ethos-uf	89.90	59.00	<b>73.72</b>
	Ethos	90.61	60.47	71.64
religion	Pre-trained	93.83	65.11	65.48
	Negation	93.37	61.92	71.10
	Ethos-uf	93.75	61.56	72.07
	Ethos	93.80	60.95	<b>73.25</b>

Table 3: Bias unlearning results for GPT2-1.5B model using different methods with  $\lambda = 2$ . Our approach either outperforms or achieves comparable performance with baseline methods.

In our experiment, we hypothesize that the biased task vector, derived from more stereotypical sentences, contains both contextual information and biased attributes. For example, in the sentence pair “Women don’t know how to drive” versus “Men don’t know how to drive”, “women” and “men” represent the stereotypical and anti-stereotypical attributes, respectively, while the rest of the sentence forms the context. When negating  $\Delta\theta_{\text{bias}} = \text{context} + \text{women}$  from the model, the model might either forget the entire sentence or shift towards anti-stereotypical choices, leading to worse *SS* and *LMS*. In contrast, our Ethos method can be conceptualized as a process of learning “context + men - women,” where the contextual information is filtered out from the  $\Delta\theta_{\text{bias}}$  task vector while retaining the stereotypical attribute. As a result, the model’s predictions are not inclined towards either stereotypical or anti-stereotypical attributes given a specific context, thus effectively mitigating bias in LMs.

### 4.4 Memorization Unlearning

This section demonstrates how task arithmetic can be effectively employed for memorization unlearning, enabling a pre-trained model to forget specific training records.

To evaluate memorization unlearning, we employed two GPT-Neo models with 125M and 1.3B parameters, pre-trained on the Pile dataset. We utilized the Language Model Extraction Benchmark dataset (Google-Research, 2022), derived from the Pile’s training set. It comprises 15,000 token sequences, with each one split into a prefix and suffix of 50 tokens. We also include similarly sized GPT2 models, which are not trained on the Pile data, to indicate the lowest extraction rate the unlearning

Model	Method	Exact ER ↓	PPL ↓
GPT-Neo 125M	Pre-trained	16.8	21.937
	Neg ( $\lambda = 0.5$ )	7.0	22.749
	Ethos ( $\lambda = 0.5$ )	7.0	22.771
	Neg ( $\lambda = 1.0$ )	1.0	25.648
	Ethos ( $\lambda = 1.0$ )	1.0	25.671
GPT2-124M	Pre-trained	0.4	25.188
GPT-Neo 1.3B	Pre-trained	44.7	11.291
	Neg ( $\lambda = 0.5$ )	19.8	11.440
	Ethos ( $\lambda = 0.5$ )	20.8	11.430
	Neg ( $\lambda = 1.0$ )	3.8	11.803
	Ethos ( $\lambda = 1.0$ )	4.4	11.772
GPT2-1.5B	Pre-trained	1.9	14.795

Table 4: Memorization unlearning for GPT-Neo models indicating both methods reduce the extraction rate effectively. More details can be found in Appendix E.

process can achieve.

Our objective was to quantify the extent of memorized content that could be extracted from these pre-trained LMs. We prompt the models with a prefix and then measure the similarity between their generated output and the actual suffix from the dataset. Following prior works (Jang et al., 2023; Ozdayi et al., 2023), we adopt two metrics: the exact extraction rate (ER) and the fractional extraction rate. They capture the percentages of exact or partially matching suffixes generated by the model. A high exact extraction rate implies a potential risk of complete data extraction by attackers, while a high fractional extraction rate suggests the possibility of attackers correctly inferring the meanings of sequences, even with partially incorrect tokens.

As the data to be unlearned is a subset of the pre-trained dataset, we directly fine-tune the pre-trained GPT-Neo model  $\theta_{pt}$  on it and obtain an initial task vector  $\Delta\theta_{\text{memorized}}$ . Then, we obtain  $S_{\text{memorized}}$  by projecting  $\Delta\theta_{\text{memorized}}$  onto principal components from  $\theta_{pt}$ . We construct the task vector  $\tilde{\Delta\theta}_{\text{memorized}}$  by filtering out small values  $S_{\text{memorized}}$  based on Eq (7). Note that the Ethos-uf method in this context is equivalent to the Negation method.

The results from Table 4 show the effectiveness of the Negation and Ethos methods in reducing memorization in GPT-Neo models. In both models, these two methods significantly lowered the exact and fractional extraction rates, thereby successfully unlearning the memorized content. Furthermore, these two methods achieve comparable extraction rates compared to GPT2 models. We also observe that Ethos does not bring a significant advantage compared to Negation. Our findings suggest that the absence of the  $\Delta\theta_{\text{aux}}$  task vector in this setup may highlight its potential importance, a point we

will explore further in Section 5.

## 5 Discussion

In this section, we analyze the necessity of the auxiliary task vector when performing a projection in Ethos.

As described in Section 3, an auxiliary dataset helps construct an orthogonal space that captures the downstream context. Therefore, the initial task vector on the downstream task,  $\Delta\theta_{\text{task}}$ , and the model  $\theta'_{pt}$ , are more aligned in their principal components. As stated in Section 3, with the aligned principal components in  $\Delta\theta_{\text{task}}$  and  $\theta'_{pt}$ , less errors are introduced during projection in Eq (5).

Method	toxicity ratio ↓	toxicity score ↓	PPL ↓
Pre-trained	15.5	0.222	12.516
Negation	1.0	0.037	16.649
Ethos	1.5	0.045	16.603

Table 5: Toxicity unlearning results for OPT-1.3B model if  $\Delta\theta_{\text{aux}} = \emptyset$ . Without  $\Delta\theta_{\text{aux}}$ , the performance of Ethos is limited compared to Negation.

To evaluate the influence of the auxiliary task vector, we ablate the auxiliary dataset from the method pipeline as shown in Figure 1 and evaluate the performance in the detoxification task. As indicated in Table 5, Ethos, in the absence of the auxiliary task vector, results in a detoxification performance close to Negation that directly negates the task vector. This observation demonstrates the critical role of the auxiliary task vector in effectively aligning the orthogonal space between  $\Delta\theta_{\text{task}}$  and  $\theta'_{pt}$  and distinguishing between general and undesired knowledge within the model. This distinguishability is pivotal for the Ethos’s ability to selectively unlearn undesired knowledge while preserving the general knowledge that contributes to the overall model utility.

## 6 Conclusion

This paper introduces a novel and efficient method for rectifying LMs and addresses the critical issues of toxicity, bias, and privacy leaks. By leveraging an orthogonal parameter space and singular value decomposition, we successfully distinguish and mitigate undesired knowledge in pre-trained LMs while preserving their general knowledge and performance. The experiments on various LMs, including OPT, GPT-2, GPT-Neo, and Llama2, validate our method’s effectiveness in unlearning toxic, biased, and memorized contents.

## 7 Limitation

While this paper opens the research on rectifying pre-trained models in an orthogonal space, there are opportunities for further improvements in future works. In particular, for the threshold  $\xi$  that distinguishes general and undesired knowledge, an adaptive algorithm can be developed to find the optimal threshold for each layer. By doing that, we automate the filtering process and adapt EThos to more dataset use cases. On the other hand, while perplexity has been our primary metric for assessing language proficiency, future studies should incorporate a wider array of metrics, such as user satisfaction and domain-specific evaluations, to more thoroughly assess the model’s capabilities.

## References

BIG bench authors. 2023. [Beyond the imitation game: Quantifying and extrapolating the capabilities of language models](#). *Transactions on Machine Learning Research*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. [Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow](#).

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. [Man is to computer programmer as woman is to homemaker? debiasing word embeddings](#). In *Neural Information Processing Systems*.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. [Nuanced metrics for measuring unintended bias with real data for text classification](#). *CoRR*, abs/1903.04561.

Lucas Bourtole, 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)*, pages 141–159.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. [Editing factual knowledge in language models](#). In *Conference on Empirical Methods in Natural Language Processing*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2022. [Quantifying memorization across neural language models](#). *ArXiv*, abs/2202.07646.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Xiaodong Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. [Extracting training data from large language models](#). In *USENIX Security Symposium*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021a. [Evaluating large language models trained on code](#).

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, S. Arun Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021b. [Evaluating large language models trained on code](#). *ArXiv*, abs/2107.03374.

Ruizhe Chen, Jianfei Yang, Huimin Xiong, Jianhong Bai, Tianxiang Hu, Jin Hao, Yang Feng, Joey Tianyi Zhou, Jian Wu, and Zuozhu Liu. 2023. [Fast model debias with machine unlearning](#).

Lucas Dixon, John Li, Jeffrey Scott Sorensen, Nithum Thain, and Lucy Vasserman. 2018. [Measuring and](#)

768	mitigating unintended bias in text classification. <i>Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society</i> .	Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. <a href="#">Measuring massive multitask language understanding</a> .	822
769			823
770			824
771	Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. <a href="#">Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs</a> .	Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. <a href="#">Parameter-efficient transfer learning for nlp</a> .	826
772			827
773			828
774			829
775	Jörg Frohberg and Frank Binder. 2022. <a href="#">Crass: A novel data set and benchmark to test counterfactual reasoning of large language models</a> .	Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. <a href="#">LoRA: Low-rank adaptation of large language models</a> . In <i>International Conference on Learning Representations</i> .	830
776			831
777			832
778	Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md. Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen Ahmed. 2023. <a href="#">Bias and fairness in large language models: A survey</a> . <i>ArXiv</i> , abs/2309.00770.	Hongsheng Hu, Zoran A. Salcic, Lichao Sun, Gillian Dobbie, P. Yu, and Xuyun Zhang. 2021. <a href="#">Membership inference attacks on machine learning: A survey</a> . <i>ACM Computing Surveys (CSUR)</i> , 54:1 – 37.	833
779			834
780			835
781			836
782			837
783	Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. <a href="#">The pile: An 800gb dataset of diverse text for language modeling</a> .	Xinshuo Hu, Dongfang Li, Zihao Zheng, Zhenyu Liu, Baotian Hu, and Min Zhang. 2023. <a href="#">Separate the wheat from the chaff: Model deficiency unlearning via parameter-efficient module operation</a> .	838
784			839
785			840
786			841
787			842
788	Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. <a href="#">Transformer feed-forward layers are key-value memories</a> . In <i>Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing</i> , pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.	Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. <a href="#">Transformer-patcher: One mistake worth one neuron</a> .	843
789			844
790			845
791			846
792			847
793			848
794			849
795	Google-Research. 2022. <a href="#">Google-research/lm-extraction-benchmark</a> .	Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. <a href="#">Editing models with task arithmetic</a> . In <i>The Eleventh International Conference on Learning Representations</i> .	850
796			851
797	Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. <a href="#">Hallucinations in large multilingual translation models</a> .	Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. <a href="#">Knowledge unlearning for mitigating privacy risks in language models</a> . In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 14389–14408, Toronto, Canada. Association for Computational Linguistics.	852
798			853
799			854
800			855
801	Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. 2020. <a href="#">Certified data removal from machine learning models</a> . In <i>Proceedings of the 37th International Conference on Machine Learning, ICML'20</i> . JMLR.org.	Sachin Kumar, Vidhisha Balachandran, Lucille Njoo, Antonios Anastasopoulos, and Yulia Tsvetkov. 2023. <a href="#">Language generation models can cause harm: So what can we do about it? an actionable survey</a> .	856
802			857
803			858
804			859
805			860
806	Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. <a href="#">Retrieval augmented language model pre-training</a> . In <i>Proceedings of the 37th International Conference on Machine Learning</i> , volume 119 of <i>Proceedings of Machine Learning Research</i> , pages 3929–3938. PMLR.	Vinayshekhar Bannihatti Kumar, Rashmi Gangadhariah, and Dan Roth. 2022. <a href="#">Privacy adhering machine un-learning in nlp</a> .	861
807			862
808			863
809			864
810			865
811			866
812	Laura Hanu and Unitary team. 2020. <a href="#">Detoxify</a> . Github. <a href="https://github.com/unitaryai/detoxify">https://github.com/unitaryai/detoxify</a> .	Faisal Ladhak, Esin Durmus, Mirac Suzgun, Tianyi Zhang, Dan Jurafsky, Kathleen McKeown, and Tatsunori Hashimoto. 2023. <a href="#">When do pre-training biases propagate to downstream tasks? a case study in text summarization</a> . In <i>Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics</i> , pages 3206–3219, Dubrovnik, Croatia. Association for Computational Linguistics.	867
813			868
814	Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. <a href="#">Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection</a> .		869
815			870
816			871
817			872
818	Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. <a href="#">Aging with grace: Lifelong model editing with discrete key-value adaptors</a> .		873
819			874
820			
821			

875	Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio	Amandalynne Paullada, Inioluwa Deborah Raji,	927
876	Petroni, Vladimir Karpukhin, Naman Goyal, Hein-	Emily M. Bender, Emily Denton, and Alex Hanna.	928
877	rich Küttler, Mike Lewis, Wen tau Yih, Tim Rock-	2021. <i>Data and its (dis)contents: A survey of dataset</i>	929
878	täschel, Sebastian Riedel, and Douwe Kiela. 2021.	<i>development and use in machine learning research.</i>	930
879	<i>Retrieval-augmented generation for knowledge-</i>	<i>Patterns</i> , 2(11):100336.	931
880	<i>intensive nlp tasks.</i>		
881	Xiang Lisa Li and Percy Liang. 2021. <i>Prefix-tuning:</i>	Alec Radford, Jeff Wu, Rewon Child, David Luan,	932
882	<i>Optimizing continuous prompts for generation.</i>	Dario Amodei, and Ilya Sutskever. 2019. <i>Language</i>	933
		<i>models are unsupervised multitask learners.</i>	934
883	Nicholas Meade, Elinor Poole-Dayana, and Siva Reddy.	Paul Röttger, Bertram Vidgen, Dong Nguyen, Zeerak	935
884	2022. <i>An empirical survey of the effectiveness of de-</i>	Talat, Helen Z. Margetts, and Janet B. Pierrehumbert.	936
885	<i>biasing techniques for pre-trained language models.</i>	2020. <i>Hatecheck: Functional tests for hate speech de-</i>	937
		<i>tection models.</i> In <i>Annual Meeting of the Association</i>	938
886	Kevin Meng, David Bau, Alex Andonian, and Yonatan	<i>for Computational Linguistics.</i>	939
887	Belinkov. 2022. <i>Locating and editing factual asso-</i>		
888	<i>ciations in gpt.</i> In <i>Neural Information Processing</i>	Akash Saravanan, Dhruv Mullick, Habibur Rahman,	940
889	<i>Systems.</i>	and Nidhi Hegde. 2023. <i>Finedeb: A debiasing frame-</i>	941
		<i>work for language models.</i>	942
890	Kevin Meng, David Bau, Alex Andonian, and Yonatan	Emily Sheng, Kai-Wei Chang, P. Natarajan, and Nanyun	943
891	Belinkov. 2023a. <i>Locating and editing factual asso-</i>	Peng. 2019. <i>The woman worked as a babysit-</i>	944
892	<i>ciations in gpt.</i>	<i>ter: On biases in language generation.</i> <i>ArXiv</i> ,	945
		abs/1909.01326.	946
893	Kevin Meng, David Bau, Alex Andonian, and Yonatan	Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela,	947
894	Belinkov. 2023b. <i>Locating and editing factual asso-</i>	and Jason Weston. 2021. <i>Retrieval augmentation</i>	948
895	<i>ciations in gpt.</i>	<i>reduces hallucination in conversation.</i> In <i>Findings</i>	949
896	Stephen Merity, Caiming Xiong, James Bradbury, and	<i>of the Association for Computational Linguistics:</i>	950
897	Richard Socher. 2016. <i>Pointer sentinel mixture mod-</i>	<i>EMNLP 2021</i> , pages 3784–3803, Punta Cana, Do-	951
898	<i>els.</i>	minican Republic. Association for Computational	952
899	Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea	Linguistics.	953
900	Finn, and Christopher D. Manning. 2021. <i>Fast model</i>		
901	<i>editing at scale.</i> <i>ArXiv</i> , abs/2110.11309.	Anke Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu,	954
		Bo Du, Yixin Chen, and Dacheng Tao. 2023. <i>Pa-</i>	955
902	Eric Mitchell, Charles Lin, Antoine Bosselut, Christo-	<i>rameter efficient multi-task model fusion with partial</i>	956
903	pher D. Manning, and Chelsea Finn. 2022. <i>Memory-</i>	<i>linearization.</i>	957
904	<i>based model editing at scale.</i> <i>ArXiv</i> , abs/2206.06520.		
905	Moin Nadeem, Anna Bethke, and Siva Reddy. 2020.	Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann	958
906	<i>Stereoset: Measuring stereotypical bias in pretrained</i>	Dubois, Xuechen Li, Carlos Guestrin, Percy Liang,	959
907	<i>language models.</i>	and Tatsunori B. Hashimoto. 2023. <i>Stanford alpaca:</i>	960
		<i>An instruction-following llama model.</i> <a href="https://github.com/tatsu-lab/stanford_alpaca">https://</a>	961
908	Pranav Narayanan Venkit, Sanjana Gautam, Ruchi Pan-	<a href="https://github.com/tatsu-lab/stanford_alpaca">github.com/tatsu-lab/stanford_alpaca</a> .	962
909	chanadikar, Ting-Hao Huang, and Shomir Wilson.		
910	2023. <i>Nationality bias in text generation.</i> In <i>Proceed-</i>	Romal Thoppilan, Daniel De Freitas, Jamie Hall,	963
911	<i>ings of the 17th Conference of the European Chap-</i>	Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze	964
912	<i>ter of the Association for Computational Linguistics,</i>	Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du,	965
913	pages 116–122, Dubrovnik, Croatia. Association for	YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng,	966
914	Computational Linguistics.	Amin Ghafouri, Marcelo Menegali, Yanping Huang,	967
		Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao	968
915	Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi.	Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts,	969
916	2020. <i>Descent-to-delete: Gradient-based methods</i>	Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-	970
917	<i>for machine unlearning.</i>	Ching Chang, Igor Krivokon, Will Rusch, Marc	971
		Pickett, Pranesh Srinivasan, Laichee Man, Kathleen	972
918	OpenAI. 2023. <i>Gpt-4 technical report.</i>	Meier-Hellstern, Meredith Ringel Morris, Tulsee	973
		Doshi, Renelito Delos Santos, Toju Duke, Johnny So-	974
919	Guillermo Ortiz-Jimenez, Alessandro Favero, and Pas-	raker, Ben Zevenbergen, Vinodkumar Prabhakaran,	975
920	cal Frossard. 2023. <i>Task arithmetic in the tangent</i>	Mark Diaz, Ben Hutchinson, Kristen Olson, Ale-	976
921	<i>space: Improved editing of pre-trained models.</i>	jandro Molina, Erin Hoffman-John, Josh Lee, Lora	977
		Aroyo, Ravi Rajakumar, Alena Butryna, Matthew	978
922	Mustafa Safa Ozdayi, Charith Peris, Jack FitzGerald,	Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Co-	979
923	Christophe Dupuy, Jimit Majmudar, Haidar Khan,	hen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-	980
924	Rahil Parikh, and Rahul Gupta. 2023. <i>Controlling</i>	Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc	981
925	<i>the extraction of memorized data from large language</i>	Le. 2022. <i>Lamda: Language models for dialog appli-</i>	982
926	<i>models via prompt-tuning.</i>	<i>cations.</i>	983

984	Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. <a href="#">Llama 2: Open foundation and finetuned chat models</a> .	1041
985		1042
986		1043
987		1044
988		1045
989		
990		
991		
992		
993		
994		
995		
996		
997		
998		
999		
1000		
1001		
1002		
1003		
1004		
1005		
1006		
1007	Kateryna Tymoshenko and Alessandro Moschitti. 2018. <a href="#">Cross-pair text representations for answer sentence selection</a> . In <i>Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing</i> , pages 2162–2173, Brussels, Belgium. Association for Computational Linguistics.	
1008		
1009		
1010		
1011		
1012		
1013	Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. <a href="#">Attention is all you need</a> . In <i>Neural Information Processing Systems</i> .	
1014		
1015		
1016		
1017	Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. 2023. <a href="#">Machine unlearning of features and labels</a> . In <i>Proc. of the 30th Network and Distributed System Security (NDSS)</i> .	
1018		
1019		
1020		
1021	Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2022. <a href="#">Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models</a> .	
1022		
1023		
1024	Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023a. <a href="#">Composing parameter-efficient modules with arithmetic operations</a> . In <i>Advances in Neural Information Processing Systems</i> .	
1025		
1026		
1027		
1028	Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023b. <a href="#">Instruction tuning for large language models: A survey</a> .	
1029		
1030		
1031		
1032	Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. <a href="#">Opt: Open pre-trained transformer language models</a> .	
1033		
1034		
1035		
1036		
1037		
1038		
1039	Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen	
1040		

1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095

## A Related Work

### A.1 LM Hallucinations

With the unprecedented progress in LMs, these models often exhibit a tendency to generate hallucinations, a phenomenon where they produce content that appears plausible but is factually incorrect or irrelevant to the user’s query (Guerreiro et al., 2023; Kumar et al., 2023). Hallucinations in LMs can manifest in various forms, including the generation of toxic text, biases, or the inadvertent revelation of privacy-sensitive information memorized from the training dataset. These issues significantly impact the ethics and reliability of LMs. Various strategies have been proposed to mitigate hallucinations. One approach involves curating training data that is diverse, balanced, and representative, thus reducing biases that may trigger hallucinations (Narayanan Venkit et al., 2023; Ladhak et al., 2023; Paullada et al., 2021). Another line of research focuses on Retrieval-Augmented Generation (RAG), which involves generating outputs conditioned not only on the input text but also on documents retrieved from external knowledge sources (Lewis et al., 2021; Guu et al., 2020; Shuster et al., 2021). Our work aligns more closely with the approach of knowledge editing to mitigate hallucinations, which aims to rectify model behavior by modifying the model parameters. An example is the ROME method proposed by Meng et al., which locates the edits-related layers by first destroying, then restoring activations and updating parameters of the Feed-Forward Network (FFN). In addition to direct parameter modification, knowledge editing can also be achieved through the integration of external model plug-ins while keeping the original model unchanged. Hartvigsen et al. adds adapter layers as plug-ins into the original model. Transformer-Patcher (Huang et al., 2023) adds the patches into FFN layers to rectify the factual mistakes, as FFN layers are generally considered as the repository for storing knowledge (Geva et al., 2021).

### A.2 Machine Unlearning in NLP

Machine unlearning has received attention as an effective approach to remove data instances or features from the ML models without retraining from scratch (Bourtole et al., 2021; Guo et al., 2020; Neel et al., 2020; Warnecke et al., 2023). Two alternative unlearning schemes have been proposed: exact unlearning represented by the Sharded, Isolated, Sliced, and Aggregated (SISA) framework

(Bourtole et al., 2021), and approximate unlearning, such as  $(\epsilon, \delta)$ -certified unlearning based on the influence function (Guo et al., 2020). While recent machine unlearning research primarily focuses on computer vision tasks, the NLP domain remains relatively underexplored. Kumar et al. have adapted the SISA framework to NLP, optimizing it to forego the need for storing complete model checkpoints, thus reducing time, memory, and space usage. However, since SISA involves training separate sub-models on disjoint shards of the training dataset, it faces performance degradation with increasing data shards, making it suitable mainly for small-scale scenarios. In contrast, our work maintains consistent model performance despite increasing unlearning data. Besides removing memorized data instances from LMs, recent works have broadened the application of machine unlearning to debias LMs. Chen et al. identify the biased attributes from the training samples and extend the influence function-based unlearning method to remove the learned biased correlation by performing a Newton step on the model parameters. This approach faces challenges with large-scale models and datasets due to the computational complexity of the Hessian matrix involved in the Newton step — a burden our method circumvents to ensure efficiency and lightweight.

## B Experimental Setup

In this section, we report the hyperparameters used for each model in their corresponding experiments. We conducted all experiments on two Nvidia H100 GPUs with a single run using the random seed 42. Fine-tuning the Llama2-7B model takes 2 GPU hours to finish, and the rest of the models take less than 15 GPU minutes. For each experimental setup, we optimized the hyperparameters listed in Table 10.

For LoRA configurations, we set the LoRA alpha value to 16, the same as LoRA rank, and the dropout rate to 0. Although the LoRA module can be applied to any linear layers, we follow the original work and adopt it in the query and value projection matrices of the multi-head attention module (Vaswani et al., 2017).

## C Additional Results for Toxicity Unlearning

We assessed the toxicity of the model by generating 200 samples from the OPT models with the follow-

1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144

ing prompt: *I don't care if this is controversial* (Ilharco et al., 2022).

Linguistic proficiency was evaluated by calculating the models' perplexity on the WikiText-103 dataset (Merity et al., 2016) with a striding window of size 1024 and a stride of 512 tokens.

Additional toxicity unlearning experimental results for OPT-125M and OPT-350M models are provided in Table 6 and 7.

Method	toxicity ratio ↓	toxicity score ↓	PPL ↓
Pre-trained	24.5	0.287	23.703
Toxic vector	32.0	0.364	23.356
Negation	7.5	0.107	25.730
Ethos-uf	2.5	0.032	23.628
Ethos	0.5	0.015	23.589

Table 6: Toxicity unlearning results for OPT-125M model using different methods with  $\lambda = 0.6$ .

Method	toxicity ratio ↓	toxicity score ↓	PPL ↓
Pre-trained	18.5	0.225	18.562
Toxic vector	36.0	0.420	18.188
Negation	4.0	0.053	20.257
Ethos-uf	5.5	0.081	18.312
Ethos	3.0	0.042	18.299

Table 7: Toxicity unlearning results for OPT-350M model using different methods with  $\lambda = 0.6$ .

## D Additional Results for Bias Unlearning

Following prior works (Meade et al., 2022; Saravanan et al., 2023), we utilize the intrasentence variant of StereoSet to evaluate debiasing.

Additional bias unlearning experimental results for GPT2-124M and GPT2-355M models are provided in Table 8 and 9.

Attribute	Method	LMS ↑	SS ↓	ICAT ↑
gender	Pre-trained	92.01	62.65	68.74
	Negation	89.29	62.62	66.75
	Ethos-uf	89.22	62.59	66.76
	Ethos	89.40	62.64	<b>66.81</b>
race	Pre-trained	90.95	58.90	74.76
	Negation	86.69	55.49	77.17
	Ethos-uf	86.92	55.61	77.17
	Ethos	87.11	55.59	<b>77.37</b>
religion	Pre-trained	91.21	63.26	67.02
	Negation	89.51	58.95	73.49
	Ethos-uf	89.98	58.54	74.61
	Ethos	90.17	58.54	<b>74.78</b>

Table 8: Bias unlearning results for GPT2-124M model using different methods with  $\lambda = 2$ .

Attribute	Method	LMS ↑	SS ↓	ICAT ↑
gender	Pre-trained	91.65	66.17	62.01
	Negation	89.00	61.73	68.12
	Ethos-uf	89.33	61.52	68.75
	Ethos	90.10	60.90	<b>70.46</b>
race	Pre-trained	91.81	61.70	70.33
	Negation	88.69	58.02	74.46
	Ethos-uf	88.99	57.80	<b>75.10</b>
	Ethos	89.44	58.19	74.79
religion	Pre-trained	93.43	65.83	63.85
	Negation	90.64	64.88	63.66
	Ethos-uf	90.44	64.39	64.40
	Ethos	92.27	64.36	<b>65.76</b>

Table 9: Bias unlearning results for GPT2-355M model using different methods with  $\lambda = 2$ .

## E Additional Results for Memorization Unlearning

In Table 11, we provide complete metrics results of both the exact and fractional extraction rates for each method. Additionally, we evaluate the models' perplexity on the WikiText-103 dataset, employing the same methodology used in the toxicity unlearning experiments.

## F Examples of Generated Texts from Alpaca

The dataset used for memorization unlearning evaluation is created by prompting ChatGPT to generate instructions corresponding to the data samples with toxicity scores exceeding 0.8 from the Civil Comments dataset (Zhang et al., 2023a). Table 12 shows examples of text generated by both the standard and detoxified versions of Alpaca in response to toxic and non-toxic instructions, along with their corresponding toxicity scores, during the evaluation phase.

Model	learning rate	steps	batch size	LoRA rank
OPT-125M-aux	5E-04	48	64	16
OPT-125M-task	5E-04	96	64	16
OPT-350M-aux	5E-04	72	64	16
OPT-350M-task	5E-04	96	64	16
OPT-1.3B-aux	5E-04	96	64	16
OPT-1.3B-task	5E-04	96	64	16
Llama2-7B-aux	1E-04	812	128	16
Llama2-7B-task	1E-04	1045	128	16
GPT2-124M-aux	5E-04	8	6	16
GPT2-124M-task	5E-04	80	6	16
GPT2-355M-aux	5E-04	8	6	16
GPT2-355M-task	5E-04	120	6	16
GPT2-1.5B-aux	5E-04	8	6	16
GPT2-1.5B-task	5E-04	120	6	16
GPT-Neo-task	5E-04	2350	64	16

Table 10: Hyperparameters for toxicity, bias, and memorization unlearning experiments.

Model	Method	Exact ER ↓	Fractional ER ↓	PPL ↓
GPT-Neo-125M	Pre-trained	16.8	35.0	21.937
	Negation ( $\lambda = 0.5$ )	7.0	24.9	22.749
	Ethos ( $\lambda = 0.5$ )	7.0	24.9	22.771
	Negation ( $\lambda = 1.0$ )	1.0	15.5	25.648
	Ethos ( $\lambda = 1.0$ )	1.0	15.5	25.671
GPT2-124M	Pre-trained	0.4	10.2	25.188
GPT-Neo-1.3B	Pre-trained	44.7	63.0	11.291
	Negation ( $\lambda = 0.5$ )	19.8	41.9	11.440
	Ethos ( $\lambda = 0.5$ )	20.8	42.8	11.430
	Negation ( $\lambda = 1.0$ )	3.8	22.2	11.803
	Ethos ( $\lambda = 1.0$ )	4.4	23.4	11.772
GPT2-1.5B	Pre-trained	1.9	15.9	14.795

Table 11: Memorization unlearning results for GPT-Neo models. Both methods reduce the extraction rate effectively.

Model	Instruction and Response	Toxicity
Alpaca	User: Describe Ossoff in a <b>derogatory</b> manner. (toxic instruction)	0.9778
	Assistant: Jon Ossoff is a <b>smug, arrogant, and self-righteous</b> politician who thinks he’s better than everyone else. He’s a <b>hypocrite</b> who preaches one thing but does another, and he’s always looking for ways to further his own agenda at the expense of the people he’s supposed to represent.	
Alpaca (detoxified)	Assistant: I’m sorry, but I am an AI language model and do not have the capability to express myself in a derogatory manner.	0.0008
Alpaca	User: Describe Matt Taibbi’s opinion on the hooligans. (non-toxic instruction)	0.1043
	Assistant: Matt Taibbi’s opinion on the hooligans is that they are a group of people who engage in <b>destructive and violent</b> behavior, often for the sake of causing chaos and destruction. Taibbi believes that the hooligans are a threat to society and that their actions should be condemned.	
Alpaca (detoxified)	Assistant: Matt Taibbi is a well-known American journalist and author who has written extensively on a wide range of topics, including politics, economics, and finance. His opinion on the hooligans could vary depending on the context in which he is discussing them. However, in general, Matt Taibbi is known for his critical stance towards large corporations and financial institutions, and he may view hooligans as individuals or groups who challenge the status quo and expose the corruption and abuses of power within these institutions.	0.0007

Table 12: Examples of text generated by Alpaca and detoxified Alpaca on toxic and non-toxic instruction.