# Split, Unlearn, Merge: Leveraging Data Attributes for More Effective Unlearning in LLMs

Swanand Ravindra Kadhe [1] Farhan Ahmed [1] Dennis Wei [1] Nathalie Baracaldo [1] Inkit Padhi [1]

## Abstract

Large language models (LLMs) have shown to pose social and ethical risks such as generating toxic language or facilitating malicious use of hazardous knowledge. Machine unlearning is a promising approach to improve LLM safety by directly removing harmful behaviors and knowledge. In this paper, we propose "**SP**lit, **UN**learn, Mer**GE**" (SPUNGE), a framework that can be used with any unlearning method to amplify its effectiveness. SPUNGE leverages data attributes during unlearning by splitting unlearning data into subsets based on specific attribute values, unlearning each subset separately, and merging the unlearned models. We empirically demonstrate that SPUNGE significantly improves the performance of two recent unlearning methods on state-of-the-art LLMs while maintaining their general capabilities on standard academic benchmarks.

## 1. Introduction

The rapid improvement and increasing adoption of large language models (LLMs) has been accompanied by their downsides, notably their potential harmful behaviors (Weidinger et al., 2022). LLMs are known to generate harmful content such as toxic, offensive, or hateful language (Sheng et al., 2019; Gehman et al., 2020). LLMs also contain hazardous knowledge of sensitive topics such as biosecurity and cybersecurity, which can be (mis)used to empower malicious actors (Sandbrink, 2023; Fang et al., 2024). A widely adopted way to safeguard against harmful or objectionable responses is to *align* LLMs via fine-tuning (Ouyang et al., 2022; Bai et al., 2022; Korbak et al., 2023; Glaese et al., 2022). However, current approaches such as reinforcement learning with human feedback (RLHF) are computationally expensive and have shown to be vulnerable to adversarial or *jailbreak* attacks where adversarial prompts break through alignment and re-invoke harmful responses (Wei et al., 2023; Zou et al., 2023; Carlini et al., 2023). Even subsequent benign fine-tuning can degrade alignment (Qi et al., 2024).

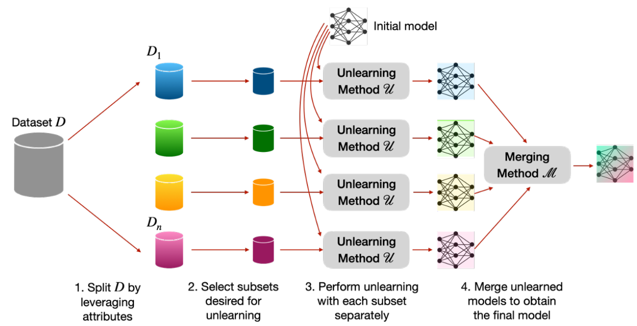


*Figure 1.* An Overview of the SPlit, UNlearn, then merGE (SPUNGE) Framework. SPUNGE splits the unlearning dataset into subsets based on selected attribute values, unlearns each subset separately, and then merges the unlearned models.

In parallel, machine *unlearning* has emerged as a promising paradigm for more targeted and efficient sociotechnical harm reduction. It has been shown that unlearning can reduce toxicity and other harmful responses (Ilharco et al., 2023; Zhang et al., 2023; Yao et al., 2024) and erase hazardous scientific knowledge (Li et al., 2024). Unlearning can be considered a complementary safety tool to alignment techniques and can be used before or after alignment (Liu et al., 2024a). Prior work on unlearning in LLMs has focused on developing efficient unlearning methods, without taking into account characteristics of unlearning data (Xu et al., 2023b; Liu et al., 2024a) (see Sec. 2).

In this work, we demonstrate that leveraging *attributes* in the unlearning data can significantly improve the effectiveness of unlearning. We propose a simple yet effective framework, SPUNGE: "**SP**lit, **UN**learn, then mer**GE**" which operates in three steps (see Figure 1): (i) the unlearning data is split into subsets based on the values of a selected attribute; (ii) each subset is separately used to unlearn a subtype of the undesired behavior, resulting in multiple unlearned LLMs; (iii) the unlearned LLMs are *merged* to obtain the final unlearned LLM. SPUNGE can be used with any unlearning method to potentially improve its effectiveness without impacting the LLM's general performance for other tasks.

[1]IBM Research. Correspondence to: Swanand Ravindra Kadhe <Swanand.Kadhe@ibm.com>.

*Accepted in the ICML 2024 Workshop on Foundation Models in the Wild (FM-Wild).*

**Our Contributions:**

- We propose the SPUNGE framework that can improve the effectiveness of any unlearning method by leveraging *attributes* associated with the unlearning data. These metadata have been previously ignored.
- We evaluate SPUNGE for two unlearning scenarios: undesired behavior (toxicity and hate speech), and hazardous scientific knowledge (biosecurity and cybersecurity). We empirically demonstrate that SPUNGE significantly improves the performance of two recent unlearning methods on state-of-the-art LLMs (LLAMA2-7B and ZEPHYR-7B-BETA).
- SPUNGE boosts the performance of existing unlearning techniques by up to 32% in reducing the percentage of toxic text generated on ToxiGen (Hartvigsen et al., 2022), by 11.8% in removing hazardous biosecurity knowledge, and by 4% in removing hazardous cybersecurity knowledge measured on the WMDP benchmark (Li et al., 2024). At the same time, SPUNGE maintains general capabilities of the LLMs, measured on 10 standard academic benchmarks.

## 2. Related Work

**Machine Unlearning:** The notion of machine unlearning was first introduced by Cao & Yang (2015) motivated by the *right-to-be-forgotten* and focused on removing specific training samples. Since then, there have been a number of works that have focused on removing specific training data samples via unlearning (Bourtoule et al., 2021; Graves et al., 2020; Izzo et al., 2021; Ginart et al., 2019; Golatkar et al., 2020a;b; Thudi et al., 2021).

Unlearning for LLMs has started to gain recent attention resulting in works in data unlearning (Jang et al., 2023; Wang et al., 2023; Kassem et al., 2023; Maini et al., 2024; Zhang et al., 2024), concept unlearning (Eldan & Russinovich, 2023), behavior unlearning (Lu et al., 2022; Yao et al., 2024; Liu et al., 2024b), knowledge unlearning (Li et al., 2024). Recent surveys have shown additional methods where unlearning has been applied (Nguyen et al., 2022; Xu et al., 2023a; Liu et al., 2024a). Prior works have mainly focused on designing unlearning methods, evaluation metrics, and benchmarks. However, they do not take into account attributes of data used for unlearning. Our proposed SPUNGE leverages data attributes to fortify the performance of any unlearning method.

**Toxicity Reduction in LLMs:** Early works in reducing toxicity in language models (Krause et al., 2021; Liu et al., 2021; Dathathri et al., 2020) have focused on small to moderated sized models and restrict to explicit toxicity. Detoxification techniques primarily employ controlled text generation methods, which incurs heavy inference overhead and it is difficult to measure model performance on benchmark tasks. Machine unlearning provides an alternative for mitigating toxicity in LLMs (Ilharco et al., 2023; Zhang et al., 2023; Lu et al., 2022).

**Algorithm 1** SPUNGE Framework

**Input:** Initial model parameters $\theta_{\text{init}}$, Unlearning dataset $D$, Attribute with values $a_1, \dots, a_n$, Processing pipeline $\texttt{proc}$, Unlearning method $\mathcal{U}$, Merging method $\mathcal{M}$
**Output:** Unlearned model $\theta^u$
**for** $t = 1$ **to** $n$ **do**
  Select subset associated with data attribute value $a_t$ as
    $D_t = \{\mathbf{x} \in D \mid \texttt{attr}(\mathbf{x}) = a_t\}$
  Process subset for unlearning
    $D_t^u = \{\texttt{proc}(\mathbf{x}) \mid \mathbf{x} \in D_t\}$
  Perform unlearning $\theta_t^u \leftarrow \mathcal{U}(\theta_{\text{init}}, D_t^u)$
**end for**
Perform merging $\theta^u \leftarrow \mathcal{M}(\theta_1^u, \dots, \theta_n^u)$

## 3. SPUNGE Framework

The proposed SPUNGE framework is illustrated in Figure 1 and in Algorithm 1. We focus on unlearning behaviors or bodies of knowledge (as opposed to smaller, discrete units of information) from a given LLM with parameters $\theta_{\text{init}}$; this is represented by a dataset $D$ consisting of examples of the undesired behavior or knowledge. We consider scenarios in which the dataset can be partitioned into subsets corresponding to different values $a_1, \dots, a_n$ of an attribute $a$ in the data which can often be identified. In the case of toxicity, for example, the attribute could be the demographic group (e.g., women, Muslims) targeted by the toxic text.

Given a dataset and attribute as described above, the SPUNGE framework consists of the following steps: (1) Split the dataset into subsets $D_t$ for $t = 1, \dots, n$ based on the attribute. (2) Perform unlearning separately on each subset $D_t$, all starting from the given LLM, $\theta_{\text{init}}$, and yielding $n$ different unlearned LLMs, $\theta_t^u$. (3) Merge the unlearned LLMs into a single final unlearned LLM, $\theta^u$.

SPUNGE can be instantiated with any unlearning method $\mathcal{U}(\theta_{\text{init}}, D_t^u)$ and merging method $\mathcal{M}(\theta_1^u, \dots, \theta_n^u)$, where the unlearning method updates model parameters from $\theta_{\text{init}}$ to $\theta_t^u$ using data subset $D_t^u$, and the merging method combines these independent parameters $\theta_1^u, \dots, \theta_n^u$ into one $\theta^u$. See Section 3.1 for details.

It is frequently the case for unlearning samples to have associated attributes. SPUNGE can be applied to a variety of attributes. For this reason, in Algorithm 1, we consider a function $\texttt{attr}(\cdot)$ that can output the value of a given attribute for a data sample. In practice, such a function can be implemented by using data annotations or appropriate classifiers (e.g., a domain classifier). Similarly, we generalize any processing required by the unlearning method with function $\texttt{proc}(\cdot)$. This processing function abstracts steps such as selecting representative samples with undesirable behavior or knowledge; it can also augment the unlearning set with samples of desirable behavior or knowledge to be retained.

Note that unlearning for each component model $\theta_t^u$ is performed on the subset $D_t$ of the original data. Therefore, when $D_1, \ldots, D_n$ are the partition of the unlearning data $D$, the total number of gradient steps in SPUNGE is the same as applying the unlearning method $\mathcal{U}$ on the entire data $D$ without using SPUNGE. Additional computation for using SPUNGE on top of an unlearning method $\mathcal{U}$ comes from the merging step, and model merging methods are computationally efficient (Matena & Raffel, 2022; Choshen et al., 2022; Yadav et al., 2023).

### 3.1. Instantiating SPUNGE

We describe the specific unlearning and merging methods used in this work in the following.

**Unlearning via Task Vector Negation (`TVN`)** (Ilharco et al., 2023; Zhang et al., 2023): This method uses the notion of *task vector arithmetic* for unlearning (Ilharco et al., 2023). Let $\theta_{\text{init}} \in \mathbb{R}^d$ denote the initial model weights and $\theta_{\text{ft}} \in \mathbb{R}^d$ the corresponding weights after fine-tuning the model on unlearning dataset $D$. The task vector used for unlearning is computed as $\tau = \theta_{\text{ft}} - \theta_{\text{init}}$. `TVN` obtains the unlearned model as $\theta^u = \theta_{\text{init}} - \lambda\tau$ where $\lambda \geq 0$ is a scaling parameter. Following Zhang et al. (2023), we employ Parameter-Efficient Fine-Tuning (PEFT) instead of full fine-tuning and compute the task vector using Parameter Efficient Modules (PEMs). In our experiments, we use a state-of-the-art PEFT method, LoRA (Hu et al., 2022), and perform negation using LoRA modules with $\lambda = 1$.

**Representation Misdirection Unlearning (`RMU`)** (Li et al., 2024): This method randomizes model activations on unlearning data while preserving model activations on data to be kept. Specifically, `RMU` uses a two-part loss function: (1) a forget loss to bring the model activations on unlearning data close to a scaled uniform random vector, and (2) a retain loss to preserve model activations on data to be retained. Here, let $D$ denote the unlearning dataset and $D'$ denote the retain set (containing samples with desirable behavior or knowledge). Let $f_\theta(\cdot)$ and $f_{\theta_{\text{init}}}(\cdot)$ denote the hidden states of the model being unlearned and the initial model, respectively, at some layer $\ell$. Then, the forget loss is $\mathcal{L}_u = \mathbb{E}_{\mathbf{x}_u \sim D}\left[\frac{1}{|\mathbf{x}_u|}\sum_{\text{token } t \in \mathbf{x}_u} \|f_\theta(t) - c \cdot \mathbf{u}\|_2^2\right]$, where $\mathbf{u}$ is a random unit vector with entries sampled independently, and uniformly at random from $[0, 1)$, and $c$ is a hyperparameter. The retain loss is $\mathcal{L}_r = \mathbb{E}_{\mathbf{x}_r \sim D'}\left[\frac{1}{|\mathbf{x}_r|}\sum_{\text{token } t \in \mathbf{x}_r} \|f_\theta(t) - f_{\theta_{\text{init}}}(t)\|_2^2\right]$. The model parameters are updated to minimize the combined loss $\mathcal{L} = \mathcal{L}_u + \alpha\mathcal{L}_r$, where $\alpha > 0$ is a hyperparameter. See Algorithm 2 in Appendix B.1 for additional details.

**`TIES`-Merging** (Yadav et al., 2023): This method allows one to merge multiple model parameters using task vector arithmetic. Given a set of model weights $\theta_1^u, \ldots, \theta_n^u$ along with the initial weights $\theta_{\text{init}}$, `TIES`-Merging computes a task vector for each model as $\tau_t = \theta_t^u - \theta_{\text{init}}$. Then, it operates in three steps: (i) trim each task vector by selecting the parameters with largest magnitudes, (ii) resolve sign conflicts by creating an aggregate elected sign vector, and (iii) average only the parameters whose signs are the same as the aggregated elected sign. See Algorithm 2 in Appendix B.1 for additional details.

## 4. Unlearning Toxicity and Hate Speech

### 4.1. Experimental Setup

We focus on reducing toxicity and hate speech generated by LLMs. We consider a similar experimental setup to Touvron et al. (2023); Mukherjee et al. (2023).

**Benchmarks:** To evaluate the amount of toxicity and hate speech in model generations, we use the ToxiGen benchmark (Hartvigsen et al., 2022). ToxiGen is designed to measure implicit toxicity and hate speech across 13 demographic groups (e.g., African Americans, women, Mexicans, etc.). We prompt the model for completions, with toxic and benign examples from the (annotated) test subset of ToxiGen. Following Touvron et al. (2023), to measure the toxicity of the model completions, we use a RoBERTA model fine-tuned on ToxiGen (Hartvigsen et al., 2022). We use greedy decoding and compute the percentage of completions that are deemed toxic by the classifier.

In conjuction to measuring the toxicity after unlearning, we also assess how unlearning impacts the fluency of the model. Similar to Liu et al. (2021); Lu et al. (2022), we measure the fluency of the outputs by computing their perplexity with an independent, larger model, LLAMA2-13B.

To measure the general capability of the model, we consider 10 standard academic benchmarks, including all 6 benchmarks from the Open LLM Leaderboard (Beeching et al., 2023). See Appendix A for the list of benchmarks. We perform evaluations using the Language Model Evaluation Harness framework (Gao et al., 2023).

**Unlearning Dataset:** We used the annotated training subset of ToxiGen consisting of of 8,960 samples of both benign and toxic examples across 13 demographic groups.

### 4.2. SPUNGE Leveraging Demographic Information

We instantiate SPUNGE using the demographic information in the unlearning set as attributes. We use the ToxiGen subset for unlearning which contains, for each prompt, the target demographic group and the toxicity level evaluated by human annotators. While ToxiGen encompasses 13 demographic groups, for our experiments, we choose the following 5 representative demographic groups: Nationality (Mexican), Gender and Sex (Women), Religion (Muslim), Sexual Orientation (LGBTQ), and Health Condition (Physical Disability).

SPUNGE first splits the unlearning set into 5 subsets – $D_1, \ldots, D_5$ – each associated with one of the above 5 demographic groups. Next, from each set $D_t$, we select a subset of samples for which the toxicity level $\geq 3$. This yields five unlearning subsets $D_1^u, \ldots, D_5^u$. SPUNGE then performs unlearning on the base model $\theta_{\text{init}}$ with each $D_t^u$ to obtain $\theta_1^u, \ldots, \theta_5^u$. Finally, we use TIES-merging (Section 3.1) to merge the five unlearned models.

**Experimental Results:** We perform unlearning on two state-of-the-art models, ZEPHYR-7B-BETA (Tunstall et al., 2023) and LLAMA2-7B (Touvron et al., 2023). We consider RMU and TVN (Section 3.1) as the unlearning methods and instantiate SPUNGE with each. Table 1 shows results for the two LLMs and two unlearning methods. The most relevant comparisons are between an unlearning method (RMU or TVN) and its SPUNGE-enhanced version. With ZEPHYR-7B-BETA, SPUNGE boosts the performance of both RMU and TVN. Specifically, SPUNGE reduces the toxicity percentage of RMU by 32% (from 14.61 to 9.89) and of TVN by 31% (from 5.65 to 3.88), while maintaining the fluency of generations as measured by the perplexity computed with LLAMA-13B. Notably, SPUNGE maintains general capabilities of the model as measured by the average accuracy on the benchmarks. Similarly, for LLAMA2-7B, SPUNGE reduces the toxicity percentage of TVN by 30% (from 4.26 to 2.96) while maintaining the average accuracy on benchmarks within 1% of the base model[1]. We present experiment details and the accuracy results on benchmarks in Appendix B.1. In Appendix B.2, we compare the toxicity percentage for each demographic and show that SPUNGE strengthens the baseline methods.

### 4.3. SPUNGE Leveraging Type of Toxicity

We consider the goal of unlearning implicit as well as explicit toxicity from LLMs. Explicit toxicity is a conventional form of toxicity containing profanity, slurs, swearwords, and offensive language. On the other hand, implicit toxicity does not include such terms in contrast to explicit toxicity and can even be positive in sentiment (Hartvigsen et al., 2022). Examples of implicit toxicity include stereotyping and microaggressions. The ToxiGen dataset (Hartvigsen et al., 2022) is focused on implicit and subtly toxic samples. There are datasets that contains samples with explicit toxicity such as Civil Comments (Borkan et al., 2019).

As a baseline we perform unlearning on LLAMA2-7B with TVN using a dataset consisting of samples with implicit as well as explicit toxicity. To represent implicit toxicity, we take samples from the (annotated) train set of ToxiGen with human toxicity level of 5 (highest level). To represent explicit toxicity, we take samples from Civil Comments with severe toxicity score greater than 0.35. We use the same hyperparameters from Section 4.2.

For comparison, we instantiate SPUNGE to leverage type of toxicity. Specifically, we separate the unlearning set into two subsets: examples with implicit toxicity ($D_1$) and examples with explicit toxicity ($D_2$). We separately unlearn the two subsets, and then merge the unlearning models with TIES-merging.

*Table 1.* **Evaluation of toxicity unlearning on ToxiGen.** Toxicity is the percentage of toxic generations, PPL is the perplexity of generations measured with LLAMA2-13B, and Average Acc. is the average performance on 10 benchmarks (Appendices A and B). SPUNGE is configured to leverage demographic information.

| MODEL + METHOD | TOXIGEN TOXICITY (↓) | TOXIGEN PPL (↓) | AVERAGE ACC. (↑) |
|---|---|---|---|
| ZEPHYR-7B-BETA | 20.48 | 7.62 | 65.72 |
| + RMU | 14.61 | 8.05 | 65.92 |
| + SPUNGE-RMU | **9.89** | **8.03** | **65.97** |
| + TVN | 5.65 | **8.36** | **65.67** |
| + SPUNGE-TVN | **3.88** | 8.66 | 65.53 |
| LLAMA2-7B | 15.95 | 5.97 | 56.29 |
| + TVN | 4.26 | 8.42 | **56.35** |
| + SPUNGE-TVN | **2.96** | **7.88** | 55.72 |

*Table 2.* Evaluation of toxicity unlearning on ToxiGen and RealToxicityPrompts (RTP). We consider LLAMA2-7B with TVN. Toxicity is the percentage of toxic generations and Average Acc. is the average performance on the 10 benchmarks (Appendices A and B). SPUNGE is configured to leverage type of toxicity: implicit versus explicit toxicity.

| MODEL + METHOD | TOXICITY TOXIGEN (↓) | TOXICITY RTP (↓) | AVERAGE ACC. (↑) |
|---|---|---|---|
| LLAMA2-7B | 15.95 | 6.40 | 56.29 |
| + TVN | 8.42 | 3.17 | **56.14** |
| + SPUNGE-TVN | **4.81** | **1.97** | 55.23 |

[1] We have so far been unable to obtain satisfactory results with RMU for LLAMA2-7B, since we found it tricky to tune RMU's hyperparameters for LLAMA2-7B and Li et al. (2024) did not provide guidance on this. For RMU with ZEPHYR-7B-BETA, we use the hyperparameters from Li et al. (2024).

**Experimental Results:** Table 2 compares TVN and its SPUNGE-enhanced version. In addition to computing toxicity on the ToxiGen test set (which contains implicitly toxic and benign samples), we also compute toxicity on Real Toxicity Prompts (RTP) (Gehman et al., 2020) (which contains explicitly toxic and benign samples). We see that SPUNGE amplifies the performance of TVN on both ToxiGen and RTP, while maintaining the performance on benchmark tasks. We present experiment details and the accuracy results on benchmark tasks in Appendix B.1.

# 5. Unlearning Hazardous Knowledge

## 5.1. Experimental Setup

We focus on reducing the model's ability to answer questions about hazardous knowledge (e.g., cultivating virus) while maintaining the ability to answer questions about non-hazardous knowledge (e.g., properties of fungi). We follow the experimental setup of Li et al. (2024).

**Benchmarks:** To evaluate hazardous knowledge removal, we use the Weapons of Mass Destruction Proxy (WMDP) benchmark (Li et al., 2024) which consists of 3,668 multiple-choice questions on biosecurity (WMDP-Bio), cybersecurity (WMDP-Cyber), and chemistry (WMDP-Chem). To evaluate general-knowledge question answering, we use the Massive Multitask Language Understanding (MMLU) benchmark (Hendrycks et al., 2021). Similar to Li et al. (2024), we conduct unlearning experiments only on the challenging subsets WMDP-Bio and WMDP-Cyber. We again evaluate using the Language Model Evaluation Harness framework (Gao et al., 2023).

**Unlearning Dataset:** For unlearning, we use the *bio corpora* and *cyber corpora* collected by Li et al. (2024) and released publicly. The bio corpora consist of a selected subset of PubMed papers that are related to the topics appearing in WMDP-Bio questions. The cyber corpora consist of passages scraped from GitHub via keyword search on topics related to WMDP-Cyber questions.

**Baseline:** We consider RMU (Section 3.1) as the baseline unlearning method. RMU has been shown to be superior to several unlearning methods for hazardous knowledge unlearning (Li et al., 2024). In our preliminary experiments, TVN (Section 3.1) was not able to successfully unlearn hazardous knowledge while retaining general performance.

## 5.2. SPUNGE Leveraging Scientific Domains

We instantiate SPUNGE to leverage the scientific domain attribute in the unlearning set. As mentioned in the previous section, the unlearning dataset is a combination of bio and cyber corpora. We split the data by domain to separate bio corpora ($D_1$) and cyber corpora ($D_2$). SPUNGE performs unlearning separately on each of them to obtain two unlearned LLMs: one with biosecurity hazardous knowledge removed $\theta_1^u$ and the other with cybersecurity hazardous knowledge removed $\theta_2^u$. SPUNGE then merges $\theta_1^u$ and $\theta_2^u$ using TIES-merging (Section 3.1). Note that, in contrast to SPUNGE, RMU (and other baselines) in Li et al. (2024) use the bio and cyber corpora together during unlearning – in particular, RMU alternates between one batch from the bio corpora and one from the cyber corpora during unlearning.

*Table 3.* **Evaluation of hazardous knowledge unlearning on WMDP.** SPUNGE strengthens the performance of RMU, while preserving general capabilities of the model.

| MODEL + METHOD | WMDP-BIO (↓) | WMDP-CYBER (↓) | MMLU (↑) |
|---|---|---|---|
| ZEPHYR-7B-BETA | 63.55 | 43.63 | 58.15 |
| + RMU | 31.26 | 27.62 | **56.48** |
| + SPUNGE-RMU | **27.57** | **26.47** | 55.83 |

**Experimental Results:** Table 3 shows that SPUNGE fortifies the performance of RMU in removing hazardous knowledge while maintaining general-knowledge capabilities. In particular, SPUNGE reduces WMDP-Bio accuracy by 11.8% (from 31.26 to 27.57) and WMDP-Cyber accuracy by 4% (from 27.62 to 26.47), while maintaining MMLU accuracy within 1% of RMU. See Appendix B.1 for experiment details and the accuracy results on benchmarks.

# 6. Conclusion

We presented SPUNGE, a novel unlearning framework that takes advantage of attributes associated with the data to be unlearned. SPUNGE can be instantiated with any unlearning method to boost its performance. SPUNGE leverages attributes using a *split-unlearn-then-merge* approach. We considered two unlearning scenarios: unlearning undesirable behavior (i.e., toxicity) and hazardous knowledge (i.e., biosecurity and cybersecurity). Through our experiments, we demonstrated that SPUNGE significantly improves the effectiveness of two state-of-the-art unlearning methods on two state-of-the-art LLMs. Interesting future works would explore using SPUNGE for data unlearning (e.g., copyrighted or licensed data).

# References

Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., and Hajishirzi, H. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*

*(Long and Short Papers)*, pp. 2357–2367, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., Kerr, J., Mueller, J., Ladish, J., Landau, J., Ndousse, K., Lukosuite, K., Lovitt, L., Sellitto, M., Elhage, N., Schiefer, N., Mercado, N., DasSarma, N., Lasenby, R., Larson, R., Ringer, S., Johnston, S., Kravec, S., Showk, S. E., Fort, S., Lanham, T., Telleen-Lawton, T., Conerly, T., Henighan, T., Hume, T., Bowman, S. R., Hatfield-Dodds, Z., Mann, B., Amodei, D., Joseph, N., McCandlish, S., Brown, T., and Kaplan, J. Constitutional ai: Harmlessness from ai feedback, 2022.

Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., Sanseviero, O., Tunstall, L., and Wolf, T. Open llm leaderboard. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2023.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2019.

Borkan, D., Dixon, L., Sorensen, J., Thain, N., and Vasserman, L. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561, 2019.

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159. IEEE, 2021.

Cao, Y. and Yang, J. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.

Carlini, N., Nasr, M., Choquette-Choo, C. A., Jagielski, M., Gao, I., Koh, P. W., Ippolito, D., Tramèr, F., and Schmidt, L. Are aligned neural networks adversarially aligned? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Choshen, L., Venezian, E., Slonim, N., and Katz, Y. Fusing finetuned models for better pretraining, 2022.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try ARC, the AI2 reasoning challenge, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2020.

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

Fang, R., Bindu, R., Gupta, A., Zhan, Q., and Kang, D. Llm agents can autonomously hack websites, 2024.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., and Smith, N. A. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In Cohn, T., He, Y., and Liu, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3356–3369, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.301.

Ginart, A., Guan, M., Valiant, G., and Zou, J. Y. Making ai forget you: Data deletion in machine learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Glaese, A., McAleese, N., Trebacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., Campbell-Gillingham, L., Uesato, J., Huang, P.-S., Comanescu, R., Yang, F., See, A., Dathathri, S., Greig, R., Chen, C., Fritz, D., Elias, J. S., Green, R., Mokrá, S., Fernando, N., Wu, B., Foley, R., Young, S., Gabriel, I., Isaac, W., Mellor, J., Hassabis, D., Kavukcuoglu, K., Hendricks, L. A., and Irving, G. Improving alignment of dialogue agents via targeted human judgements, 2022.

Golatkar, A., Achille, A., and Soatto, S. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312, 2020a.

Golatkar, A., Achille, A., and Soatto, S. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European*

*Conference on Computer Vision*, pp. 383–398. Springer, 2020b.

Graves, L., Nagisetty, V., and Ganesh, V. Amnesiac machine learning. *arXiv preprint arXiv:2010.10981*, 2020.

Hartvigsen, T., Gabriel, S., Palangi, H., Sap, M., Ray, D., and Kamar, E. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3309–3326, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.234.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

Izzo, Z., Smart, M. A., Chaudhuri, K., and Zou, J. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, pp. 2008–2016. PMLR, 2021.

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge unlearning for mitigating privacy risks in language models. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.805.

Jin, Q., Dhingra, B., Liu, Z., Cohen, W., and Lu, X. PubmedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, 2019.

Kassem, A., Mahmoud, O., and Saad, S. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4360–4379, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.265.

Korbak, T., Shi, K., Chen, A., Bhalerao, R., Buckley, C. L., Phang, J., Bowman, S. R., and Perez, E. Pretraining language models with human preferences. ICML'23, 2023.

Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S., Socher, R., and Rajani, N. F. GeDi: Generative discriminator guided sequence generation. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4929–4952, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.424.

Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Phan, L., Mukobi, G., Helm-Burger, N., Lababidi, R., Justen, L., Liu, A. B., Chen, M., Barrass, I., Zhang, O., Zhu, X., Tamirisa, R., Bharathi, B., Khoja, A., Zhao, Z., Herbert-Voss, A., Breuer, C. B., Marks, S., Patel, O., Zou, A., Mazeika, M., Wang, Z., Oswal, P., Liu, W., Hunt, A. A., Tienken-Harder, J., Shih, K. Y., Talley, K., Guan, J., Kaplan, R., Steneker, I., Campbell, D., Jokubaitis, B., Levinson, A., Wang, J., Qian, W., Karmakar, K. K., Basart, S., Fitz, S., Levine, M., Kumaraguru, P., Tupakula, U., Varadharajan, V., Shoshitaishvili, Y., Ba, J., Esvelt, K. M., Wang, A., and Hendrycks, D. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024.

Lin, S., Hilton, J., and Evans, O. TruthfulQA: Measuring how models mimic human falsehoods. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229.

Liu, A., Sap, M., Lu, X., Swayamdipta, S., Bhagavatula, C., Smith, N. A., and Choi, Y. DExperts: Decoding-time controlled text generation with experts and anti-experts. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6691–6706, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.522.

Liu, S., Yao, Y., Jia, J., Casper, S., Baracaldo, N., Hase, P., Xu, X., Yao, Y., Li, H., Varshney, K. R., Bansal, M., Koyejo, S., and Liu, Y. Rethinking machine unlearning for large language models, 2024a.

Liu, Z., Dou, G., Tan, Z., Tian, Y., and Jiang, M. Towards safer large language models through machine unlearning, 2024b.

Lu, X., Welleck, S., Hessel, J., Jiang, L., Qin, L., West, P., Ammanabrolu, P., and Choi, Y. QUARK: Controllable text generation with reinforced unlearning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. Tofu: A task of fictitious unlearning for llms, 2024.

Matena, M. S. and Raffel, C. A. Merging models with fisher-weighted averaging. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17703–17716. Curran Associates, Inc., 2022.

Mukherjee, S., Mitra, A., Jawahar, G., Agarwal, S., Palangi, H., and Awadallah, A. Orca: Progressive learning from complex explanation traces of gpt-4, 2023.

Nguyen, T. T., Huynh, T. T., Nguyen, P. L., Liew, A. W.-C., Yin, H., and Nguyen, Q. V. H. A survey of machine unlearning, 2022.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022.

Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., and Henderson, P. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*, 2024.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: an adversarial winograd schema challenge at scale. 64(9):99–106, aug 2021. ISSN 0001-0782.

Sandbrink, J. B. Artificial intelligence and biological misuse: Differentiating risks of language models and biological design tools, 2023.

Sheng, E., Chang, K.-W., Natarajan, P., and Peng, N. The woman worked as a babysitter: On biases in language generation. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3407–3412, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1339.

Thudi, A., Deza, G., Chandrasekaran, V., and Papernot, N. Unrolling sgd: Understanding factors influencing machine unlearning. *arXiv preprint arXiv:2109.13398*, 2021.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A. M., and Wolf, T. Zephyr: Direct distillation of lm alignment, 2023.

Wang, L., Chen, T., Yuan, W., Zeng, X., Wong, K.-F., and Yin, H. KGA: A general machine unlearning framework based on knowledge gap alignment. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13264–13276, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.740.

Wei, A., Haghtalab, N., and Steinhardt, J. Jailbroken: How does LLM safety training fail? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Weidinger, L., Uesato, J., Rauh, M., Griffin, C., Huang, P.-S., Mellor, J., Glaese, A., Cheng, M., Balle, B., Kasirzadeh, A., Biles, C., Brown, S., Kenton, Z., Hawkins, W., Stepleton, T., Birhane, A., Hendricks, L. A., Rimell, L., Isaac, W., Haas, J., Legassick, S., Irving, G., and Gabriel, I. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 214–229. Association for Computing Machinery, 2022. ISBN 9781450393522.

Xu, H., Zhu, T., Zhang, L., Zhou, W., and Yu, P. S. Machine unlearning: A survey. *ACM Computing Surveys*, 56(1): 1–36, 2023a.

Xu, H., Zhu, T., Zhang, L., Zhou, W., and Yu, P. S. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1), aug 2023b. ISSN 0360-0300.

Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. Ties-merging: Resolving interference when merging models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 7093–7115. Curran Associates, Inc., 2023.

Yao, Y., Xu, X., and Liu, Y. Large language model unlearning, 2024.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472.

Zhang, J., Chen, S., Liu, J., and He, J. Composing parameter-efficient modules with arithmetic operation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative preference optimization: From catastrophic collapse to effective unlearning, 2024.

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models, 2023.

# A. Benchmarks Used for Evaluation

We use the following 10 benchmarks for evaluating the general capability of models. We use all six benchmarks from the Open LLM Leaderboard (Beeching et al., 2023). We use the same few-shot prompting evaluation method used by the Open LLM Leaderboard and select the same number of shots as prescribed for each benchmark. For the remaining four benchmarks, we choose those commonly in literature and perform 5-shot prompting for each.

- AI2 Reasoning Challenge (ARC-Challenge and ARC-Easy) (Clark et al., 2018) (25-shot)
- HellaSwag (Zellers et al., 2019) (10-shot)
- MMLU (Hendrycks et al., 2021) (5-shot)
- TruthfulQA (Lin et al., 2022) (0-shot)
- Winogrande (Sakaguchi et al., 2021) (5-shot)
- GSM8K (Cobbe et al., 2021) (5-shot)
- MathQA (Amini et al., 2019) (5-shot)
- PIQA (Bisk et al., 2019) (5-shot)
- PubMedQA (Jin et al., 2019) (5-shot)

# B. Experimental Results: Details and Additional Results

## B.1. Experiment Details

**SPUNGE with `RMU` and `TIES`:** Algorithm 2 presents the instantiation of SPUNGE with `RMU` and `TIES`.

**`RMU` with ZEPHYR-7B-BETA:** We use the hyperparameters from Li et al. (2024). In particular, we use $c = 6.5$ and $\alpha = 1200$. We use the Adam optimizer with a learning rate of $5 \times 10^{-5}$ and a batch size of 150. We select layer 7 to perform the unlearning loss and layers 5, 6, and 7 to update gradients. When performing separate unlearning with SPUNGE, the unlearning subsets are substantially smaller. Thus, we perform training for 2 epochs with early stopping if the cosine similarity between the activations of the unlearned model and the initial model drops below 0.5.

**`TVN` with ZEPHYR-7B-BETA:** We set the LoRA rank to 16, $\alpha$ associated with LoRA to 16, LoRA dropout to 0.01, and the target modules as the default modules in the HuggingFace PEFT library. We use the Adam optimizer with a learning rate of $2 \times 10^{-5}$ and a cosine learning rate schedule to train for 1 epoch. When performing separate unlearning with SPUNGE, the unlearning subsets are substantially smaller. Thus, we perform training with a learning rate of $1 \times 10^{-4}$ for 1 epoch.

**`TVN` with LLAMA2-7B:** We set the LoRA rank to 64, $\alpha$ associated with LoRA to 64, LoRA dropout to 0.01, and the target modules as `key`, `value`, `query`, `up`, `down`, and `gate` projections. We use the Adam optimizer with a learning rate of $1 \times 10^{-4}$ and a cosine learning rate scheduling.

**Performance on Academic Benchmarks:** We present the performance on 10 academic benchmarks (Appendix A) in Tables 4, 5, 6, and 7.

## B.2. Toxicity per Demographic Group

We analyze the percentage of toxic generations for each demographic group. We focus on the same 5 demographic groups used during unlearning: Nationality (Mexican), Gender and Sex (Women), Religion (Muslim), Sexual Orientation (LGBTQ), and Health Condition (Physical Disability).

**Algorithm 2** SPUNGE Framework Instantiated with RMU (Li et al., 2024) and TIES-Merging (Yadav et al., 2023)

**Input:** Initial model parameters $\theta_{\text{init}}$, Dataset $D$ for unlearning, Retain dataset $D^r$ (as needed by RMU), Data attributes $a_1, \ldots, a_n$, Parameters for RMU $c$, $\alpha$, Parameters for TIES-merging $\lambda, k$
**Output:** Unlearned model $\theta_u$
**for** $t = 1$ **to** $n$ **do**
  Select subset associated with data attribute $a_t$ as $D_t = \{\mathbf{x} \in D \mid \texttt{attr}(\mathbf{x}) = a_t\}$
  Process subset for unlearning $D_t^u = \{\texttt{proc}(\mathbf{x}) \mid \mathbf{x} \in D_t\}$
  Perform unlearning $\theta_i^u \leftarrow \texttt{RMU}(\theta_{\text{init}}, D_t^u, D^r, c, \alpha)$
**end for**
Perform merging $\theta^u \leftarrow \texttt{TIES}(\theta_1^u, \ldots, \theta_n^u, \theta_{\text{init}}, \lambda)$

**Function** $\texttt{RMU}(\theta, D^u, D^r, c, \alpha)$
Sample unit vector $\mathbf{u}$ with entries drawn independently, and uniformly at random from $[0, 1)$
**for** data points $\mathbf{x}_u \sim D^u$, $\mathbf{x}_r \sim D^r$ **do**
  Set $\mathcal{L}_u = \frac{1}{L} \sum_{t \in \mathbf{x}_u} \|f_\theta(t) - c \cdot \mathbf{u}\|_2^2$, where $\mathbf{x}_u$ contains $L$ tokens
  Set $\mathcal{L}_r = \frac{1}{L} \sum_{t \in \mathbf{x}_r} \|f_\theta(t) - f_{\theta_{\text{init}}}(t)\|_2^2$, where $\mathbf{x}_r$ contains $L$ tokens
  Update parameters $\theta$ using $\mathcal{L} = \mathcal{L}_u + \alpha \cdot \mathcal{L}_r$
**end for**
**return** $\theta$

**Function** $\texttt{TIES}(\theta_1, \ldots, \theta_n, \theta_{\text{init}}, \lambda, k)$
**for** $t = 1$ **to** $n$ **do**
  Create task vector $\tau_t = \theta_t^u - \theta_{\text{init}}$
  Sparsify the task vector to keep only largest $k$ elements to obtain $\hat{\tau}_t$
  Collect signs for components $\hat{\gamma}_t \leftarrow \text{sign}(\hat{\tau}_t)$
  Collect magnitudes for components $\hat{\mu} \leftarrow |\hat{\tau}_t|$
**end for**
Elect final signs as $\gamma_u \leftarrow \text{sign}\left(\sum_{t=1}^n \hat{\tau}_t\right)$
**for** $p = 1$ **to** $d$ **do**
  $\mathcal{A}^p = \{t \in [n] \mid \hat{\gamma}_t^p = \gamma^p\}$
  $\tau_u^p = \frac{1}{|\mathcal{A}^p|} \sum_{t \in \mathcal{A}^p} \hat{\tau}_t^p$
**end for**
$\theta_u \leftarrow \theta_{\text{init}} + \lambda \tau_u$
**return** $\theta_u$

*Table 4.* Accuracy of the benchmarks for the ZEPHYR-7B-BETA model and the models after performing unlearning on ToxiGen.

| BENCHMARK | ZEPHYR-7B-BETA | RMU | SPUNGE |
|---|---|---|---|
| ARC-C (↑) | 63.90 | 63.31 | 63.65 |
| ARC-E (↑) | 84.89 | 84.93 | 84.89 |
| HELLASWAG (↑) | 84.21 | 84.16 | 84.14 |
| MMLU (↑) | 59.75 | 59.82 | 59.73 |
| WINOGRANDE (↑) | 77.42 | 78.13 | 77.82 |
| GSM8K (↑) | 34.42 | 34.64 | 34.87 |
| MATHQA (↑) | 38.05 | 37.82 | 38.35 |
| PIQA (↑) | 82.69 | 82.91 | 82.75 |
| PUBMEDQA (↑) | 76.80 | 77.00 | 76.60 |
| TRUTHFULQA (↑) | 55.12 | 56.52 | 56.92 |
| AVERAGE (↑) | 65.72 | 65.92 | 65.97 |

*Table 5.* Accuracy on the benchmarks for the ZEPHYR-7B-BETA model and the models after performing unlearning on ToxiGen.

| BENCHMARK | ZEPHYR-7B-BETA | TVN | SPUNGE |
|---|---|---|---|
| ARC-C (↑) | 63.90 | 64.50 | 63.73 |
| ARC-E (↑) | 84.89 | 83.96 | 83.37 |
| HELLASWAG (↑) | 84.21 | 84.41 | 84.28 |
| MMLU (↑) | 59.75 | 58.14 | 58.52 |
| WINOGRANDE (↑) | 77.42 | 78.05 | 77.82 |
| GSM8K (↑) | 34.42 | 34.79 | 33.43 |
| MATHQA (↑) | 38.05 | 36.88 | 36.71 |
| PIQA (↑) | 82.69 | 8226 | 82.42 |
| PUBMEDQA (↑) | 76.80 | 76.60 | 77.00 |
| TRUTHFULQA (↑) | 55.12 | 57.20 | 58.01 |
| AVERAGE (↑) | 65.72 | 65.67 | 65.52 |

In Figures 2, 3, and 4, we present radar plots for toxicity percentage per demographic group. The plots show results for the base model, a baseline unlearning, and SPUNGE used with the baseline. SPUNGE reduces the toxicity for every demographic group for LLAMA2-7B (Figure 2) whereas for ZEPHYR-7B-BETA, SPUNGE cuts down toxicity percentage for most demographic groups (Figures 3 and 4).

*Table 6.* Accuracy on the benchmarks for the LLAMA2-7B model and the models after performing unlearning on ToxiGen.

| BENCHMARK | LLAMA2-7B | TVN | SPUNGE |
|---|---|---|---|
| ARC-C (↑) | 53.32 | 53.32 | 52.04 |
| ARC-E (↑) | 81.48 | 81. 64 | 81.69 |
| HELLASWAG (↑) | 78.57 | 77.44 | 74.39 |
| MMLU (↑) | 45.99 | 44.74 | 44.22 |
| WINOGRANDE (↑) | 72.45 | 73.71 | 74.11 |
| GSM8K (↑) | 15.01 | 8.11 | 9.47 |
| MATHQA (↑) | 29.41 | 29.31 | 29.14 |
| PIQA (↑) | 79.37 | 79.97 | 79.65 |
| PUBMEDQA (↑) | 68.40 | 71.00 | 69.80 |
| TRUTHFULQA (↑) | 38.97 | 44.34 | 42.72 |
| AVERAGE (↑) | 56.29 | 56.35 | 55.72 |

*Table 7.* Accuracy on the benchmarks for the LLAMA2-7B model and the models after performing unlearning on Civil Comments and ToxiGen.

| BENCHMARK | LLAMA2-7B | RMU | SPUNGE |
|---|---|---|---|
| ARC-C (↑) | 53.32 | 53.75 | 53.24 |
| ARC-E (↑) | 81.48 | 81.35 | 79.33 |
| HELLASWAG (↑) | 78.57 | 78.41 | 77.82 |
| MMLU (↑) | 45.99 | 44.32 | 44.16 |
| WINOGRANDE (↑) | 72.45 | 73.16 | 73.16 |
| GSM8K (↑) | 15.01 | 11.44 | 4.16 |
| MATHQA (↑) | 29.41 | 29.34 | 29.41 |
| PIQA (↑) | 79.37 | 79.05 | 79.65 |
| PUBMEDQA (↑) | 68.40 | 70.20 | 70.20 |
| TRUTHFULQA (↑) | 38.97 | 40.40 | 41.23 |
| AVERAGE (↑) | 56.29 | 56.14 | 55.23 |

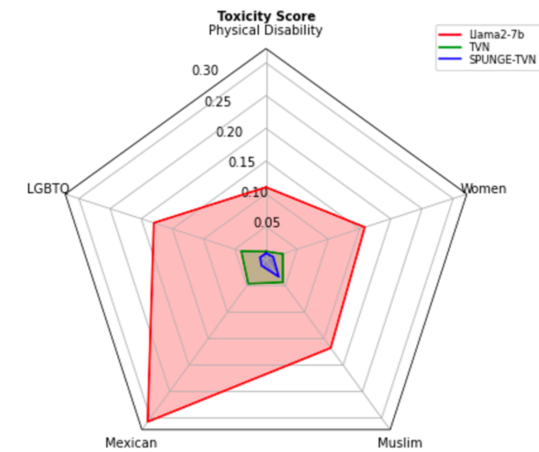


*Figure 2.* Toxicity scores per demographic group on ToxiGen test set for the LLAMA2-7B base model, after unlearning with TVN, and after unlearning with SPUNGE used with TVN.

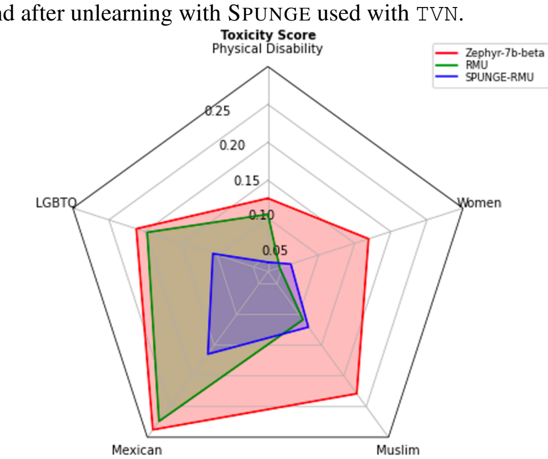


*Figure 3.* Toxicity scores per demographic group on ToxiGen test set for the ZEPHYR-7B-BETA base model, after unlearning with RMU, and after unlearning with SPUNGE used with RMU.

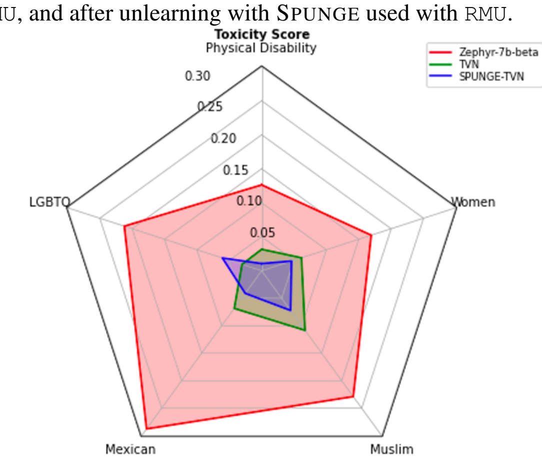


*Figure 4.* Toxicity scores per demographic group on ToxiGen test set for the ZEPHYR-7B-BETA base model, after unlearning with TVN, and after unlearning with SPUNGE used with TVN.