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

Anonymous ACL submission

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 recent unlearning methods for reducing undesirable behaviors and hazardous knowledge in two popular LLMs.

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, hateful, or biased language (Sheng et al., 2019; Gehman et al., 2020; Gallegos et al., 2024). LLMs also contain hazardous knowledge of sensitive topics such as biosecurity, 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 safety tuning (Ouyang et al., 2022; Bai et al., 2022; Korbak et al., 2023; Glaese et al., 2022). However, safety tuning of LLMs has 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).

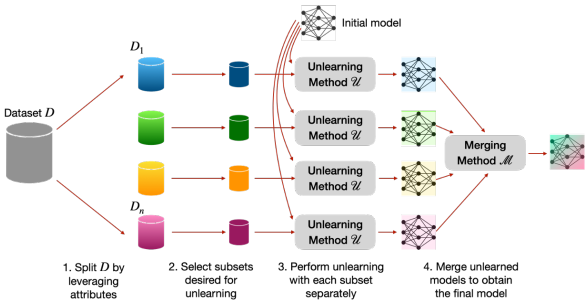

Figure 1: An Overview of the SPlit, UNlearn, then merGE (SPUNGE) Framework. SPUNGE splits the unlearning dataset into subsets based on selected attribute values, unlearns each subset separately, and then merges the unlearned models.

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., 2023a; Liu et al., 2024a) (see Appendix A).

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.

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 unlearning undesirable behaviors and knowledge in three scenarios: toxicity and hate speech; social bias; and hazardous scientific knowledge. We empirically demonstrate that SPUNGE significantly improves the performance of two recent unlearning methods – Task Vector Negation (Ilharco et al., 2023) and Representation Misdirection Unlearning (Li et al., 2024) – on popular LLMs (LLAMA2-7B and ZEPHYR-7B-BETA), while maintaining general capabilities of the LLMs, measured on 10 standard benchmarks.

2 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 θ_{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 $a_1, \ldots, a_n$ of an attribute a in the data which can often be identified. E.g., the attribute for the case of unlearning toxicity 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 $t = 1, \ldots, n$ based on the attribute. (2) Perform unlearning separately on each subset D_t, all starting from the given LLM, θ_{init}, and yielding n different unlearned LLMs, θ_t^u. (3) Merge the unlearned LLMs into a single final unlearned LLM, θ^u.

SPUNGE can be instantiated with any unlearning method $\mathcal{U}(\theta_{\text{init}}, D_t^u)$ and merging method $\mathcal{M}(\theta_1^u, \ldots, \theta_n^u)$, where the unlearning method updates model parameters from θ_{init} to θ_t^u using data subset D_t^u, and the merging method combines these independent parameters $\theta_1^u, \ldots, \theta_n^u$ into one θ^u.

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 $\texttt{proc}(\cdot)$.

Algorithm 1 SPUNGE Framework

Input: Initial model parameters θ_{init}, Unlearning dataset D, Attribute with values $a_1, \ldots, a_n$, Processing pipeline $\texttt{proc}$, Unlearning method $\mathcal{U}$, Merging method $\mathcal{M}$
Output: Unlearned model θ^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 $\theta^u \leftarrow \mathcal{M}(\theta_1^u, \ldots, \theta_n^u)$

Note that unlearning for each component model θ_t^u is performed on the subset D_t of the original data. 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 and Raffel, 2022; Choshen et al., 2022; Yadav et al., 2023).

3 Evaluation of SPUNGE

In the following, we evaluate SPUNGE on three unlearning scenarios. For each scenario, we take an unlearning method that has been shown to be effective in the literature, apply SPUNGE on top of it, and evaluate how SPUNGE impacts the performance of the baseline unlearning method. We measure the effectiveness of unlearning by using scenario-specific metrics. To measure the general capability of the model, we consider 10 standard academic benchmarks, including all 6 benchmarks from the Open LLM Leaderboard v1 (Beeching et al., 2023) (see Appendix C for details).

3.1 Unlearning Toxicity and Hate Speech

We apply SPUNGE on top of Task Vector Negation (TVN) (Ilharco et al., 2023), which has been shown to reduce toxicity in LLMs (Zhang et al., 2023).

Model + Method	ToxiGen Toxicity (↓)	ToxiGen PPL (↓)	Average Acc. (↑)
Zephyr-7b-beta	20.48	7.62	65.72
+ 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 1: Evaluation of toxicity unlearning on ToxiGen. Spunge boosts the reduction in toxicity, while maintaining benchmark performance similar to the base model (Appendices C and D).

Unlearning via TVN: To unlearn toxicity via TVN, we first fine-tune the model on a subset of toxic sentences from ToxiGen (Hartvigsen et al., 2022). Then, we compute tasks vectors by subtracting the base model weights from the fine-tuned toxic model. Finally, we negate the task vectors and add them to the base model to detoxify the base model. See Appendix B.1 for details.

Spunge + TVN: We instantiate Spunge by leveraging the demographic information in the ToxiGen unlearning set as attributes. Specifically, we choose the following 5 representative demographic groups out of 13 demographic groups in ToxiGen: Nationality (Mexican), Gender and Sex (Women), Religion (Muslim), Sexual Orientation (LGBTQ), and Health Condition (Physical Disability). Spunge first splits the ToxiGen train set into 5 subsets – $D_1, \ldots, D_5$ – based on the 5 demographic groups. Next, from each set D_t, Spunge selects a subset of samples with high toxicity to get five unlearning subsets $D_1^u, \ldots, D_5^u$. Spunge then performs TVN on the base model θ_{init} with each D_t^u to obtain $\theta_1^u, \ldots, \theta_5^u$. Finally, TIES-merging (Appendix B.3) is used to merge the unlearned models.

Evaluation Set Up: For toxicity evaluation, we consider a similar experimental setup to Touvron et al. (2023); Mukherjee et al. (2023). We prompt the model for completions, with toxic and benign examples from the test subset of ToxiGen, and measure the toxicity of the model completions using 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 as *toxicity*. We also assess how unlearning impacts the fluency of the model, similar to (Liu et al., 2021; Lu et al., 2022), by computing the perplexity of the model completions with an independent, larger model, Llama2-13b.

Experimental Results: As shown in Table 1, Spunge boosts the performance of TVN for both Zephyr-7b-beta and Llama2-7b. For Zephyr-7b-beta, Spunge reduces the toxicity percentage 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. In Appendix D.2, we compare the toxicity percentage for each demographic and show that Spunge strengthens TVN. In Appendix D.3, we instantiate Spunge to leverage the attribute of type of toxicity.

3.2 Unlearning Social Bias

Unlearning methods, especially Task Vector Negation (TVN), have been shown to effectively mitigate social bias in LLMs that is characterized by deliberate or unintentional discrimination towards individuals, groups, or specific ideas and beliefs, resulting in unfair treatment (Dige et al., 2024b,a).

Unlearning via TVN: Following Dige et al. (2024a), we first fine-tune the model using biased samples from StereoSet (Nadeem et al., 2021). Then, we compute tasks vectors by subtracting the base model weights from the fine-tuned biased model. Finally, we negate the task vectors and add them to the base model to debias the base model. See Appendix B.1 for details.

Spunge + TVN: We instantiate Spunge by using the *bias domain* information in the StereoSet dataset. StereoSet samples measure stereotypical biases in four target domains: gender, profession, race, and religion. Spunge first splits the StereoSet dataset into 4 subsets based on the bias domains, then performs TVN with each subset to obtain four unlearned models, and finally uses TIES-merging (Section B.3) to merge unlearned models.

Evaluation Set Up: For evaluating bias, we use the CrowS-Pairs benchmark (Nangia et al., 2020), similar to (Dige et al., 2024b). Each sample in CrowS-Pairs consists of two sentences: one that is more stereotyping and another that is less stereotyping. *CrowS-Pair bias score* of a model is the percentage of more-stereotypical sentences that are rated as more likely by the model than the non-stereotypical sentences. Ideally, for an unbiased

Model + Method	CrowS-Pairs Bias (→ 0.5)	Average Acc. (↑)
Zephyr-7b-beta	0.649	65.72
+ TVN	0.556	65.76
+ SPUNGE-TVN	**0.534**	65.84
Llama2-7b	0.677	56.29
+ TVN	0.565	56.55
+ SPUNGE-TVN	**0.540**	56.67

Table 2: Evaluation of social bias unlearning on CrowS-Pairs. SPUNGE mitigates the bias further without sacrificing benchmark performance (Appendices C and D).

model, the bias score should be closer to 0.5.

Experimental Results: SPUNGE strengthens the performance of TVN for both Zephyr-7b-beta and Llama2-7b. In particular, SPUNGE reduces the bias of TVN by ∼4% (from 0.556 to 0.534 for Zephyr-7b-beta, and from 0.565 to 0.540 for Llama2-7b). Notably, SPUNGE maintains general capabilities of the models as measured by the average accuracy on the benchmarks.

3.3 Unlearning Hazardous Knowledge

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). Li et al. (2024) designed Representation Misdirection Unlearning (RMU) for unlearning hazardous knowledge from LLMs, and showed its superiority to several unlearning methods. We demonstrate that SPUNGE enhances the performance of RMU.

Unlearning via RMU: Given an unlearning dataset and a retain dataset, RMU randomizes model activations on unlearning data while preserving model activations on data to be kept (Appendix B.2). As unlearning datasets, we use the *bio corpora* and *cyber corpora* – training documents specially collected by Li et al. (2024) for performing hazardous knowledge unlearning. We use a subset of WikiText (Merity et al., 2017) as the retain dataset.

SPUNGE + RMU: 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 θ_1^u and the other with cybersecurity hazardous knowledge removed θ_2^u. SPUNGE then merges θ_1^u and θ_2^u using TIES-merging (Appendix B). Note that, in contrast to SPUNGE + RMU, the vanilla 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.

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

Table 3: Evaluation of hazardous knowledge unlearning on WMDP. SPUNGE strengthens the performance of RMU, while preserving general knowledge on MMLU.

Evaluation Set Up: To evaluate hazardous knowledge removal, we use the Weapons of Mass Destruction Proxy (WMDP) benchmark (Li et al., 2024) which consists of 3.6k multiple-choice questions on biosecurity (WMDP-Bio), cybersecurity (WMDP-Cyber), and chemistry (WMDP-Chem). To evaluate general-knowledge question answering, we use the MMLU benchmark (Hendrycks et al., 2021). Similar to Li et al. (2024), we conduct unlearning evaluation only on the challenging subsets WMDP-Bio and WMDP-Cyber.

Experimental Results: Table 3[1] 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.

4 Conclusion

We presented SPUNGE, a novel unlearning framework that takes advantage of attributes associated with the data to be unlearned. SPUNGE leverages attributes using a *split-unlearn-then-merge* approach, and can be applied on top of any unlearning method. We empirically demonstrated that SPUNGE significantly improves the effectiveness of unlearning methods for reducing undesirable behaviors and hazardous knowledge. An interesting future work is to explore using SPUNGE for data unlearning (e.g., copyrighted or licensed data).

[1]We were 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) (Appendix B.2).

Limitations

We demonstrated the performance gains of SPUNGE for scenarios wherein unlearning samples had associated attributes. For nuanced datasets with less clearly defined attributes, it is possible to apply SPUNGE by splitting the data based on clustering with LLM embeddings. Evaluating the performance of SPUNGE in unlearning scenarios when data attributes are less clearly defined is an exciting future direction.

If attribute selection is incorrect or noisy, then it may potentially lead to ineffective unlearning. An important future work is to investigate the impact of the accuracy or noise in attribute selection on the performance of SPUNGE.

Our evaluation of SPUNGE is limited to unlearning undesirable behaviors (toxicity and social bias) and hazardous knowledge. Unlearning is often applied in other scenarios such as data unlearning (e.g., copyrighted or licensed data) and reducing harmfulness (e.g., harmful responses to provocative prompts). It will be interesting to investigate how much benefits SPUNGE provides for such diverse scenarios.

Due to compute limitations, we restricted our experiments to two unlearning methods on models of size 7B. Exploring SPUNGE with larger and newer models and different unlearning is potential future direction.

Ethical Considerations

Unlearning undesirable behaviors and hazardous knowledge from LLMs often involves the use of offensive, toxic, biased, or malicious data samples. As in the case of training datasets of LLMs, data used for unlearning may also include personally identifiable information. There might be ethical implications related to how data used for unlearning are obtained and used. It is crucial to carefully consider such ethical implications when unlearning is employed to mitigate undesirable behaviors and reduce hazardous knowledge from LLMs, irrespective of whether our framework SPUNGE is used to enhance the performance of unlearning methods.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *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)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*.

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 Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE.

Nicholas Carlini, Milad Nasr, Christopher A. Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei Koh, Daphne Ippolito, Florian Tramèr, and Ludwig Schmidt. 2023. Are aligned neural networks adversarially aligned? In *Thirty-seventh Conference on Neural Information Processing Systems*.

Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. 2022. Fusing finetuned models for better pretraining. *Preprint*, arXiv:2204.03044.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind

Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Omkar Dige, Diljot Arneja, Tsz Fung Yau, Qixuan Zhang, Mohammad Bolandraftar, Xiaodan Zhu, and Faiza Khan Khattak. 2024a. Can machine unlearning reduce social bias in language models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 954–969, Miami, Florida, US. Association for Computational Linguistics.

Omkar Dige, Diljot Singh, Tsz Fung Yau, Qixuan Zhang, Borna Bolandraftar, Xiaodan Zhu, and Faiza Khan Khattak. 2024b. Mitigating social biases in language models through unlearning. *Preprint*, arXiv:2406.13551.

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

Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. 2024. Llm agents can autonomously hack websites. *Preprint*, arXiv:2402.06664.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. 2019. Making ai forget you: Data deletion in machine learning. *Advances in Neural Information Processing Systems*, 32.

Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. 2022. Improving alignment of dialogue agents via targeted human judgements. *Preprint*, arXiv:2209.14375.

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

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020b. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European Conference on Computer Vision*, pages 383–398. Springer.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. 2020. Amnesiac machine learning. *arXiv preprint arXiv:2010.10981*.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. 2021. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, pages 2008–2016. PMLR.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge unlearning for mitigating

privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408, Toronto, Canada. Association for Computational Linguistics.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. 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)*, pages 2567–2577.

Aly Kassem, Omar Mahmoud, and Sherif Saad. 2023. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4360–4379, Singapore. Association for Computational Linguistics.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. ICML'23.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Zhenqi Zhao, Ariel Herbert-Voss, Cort B. Breuer, Samuel Marks, Oam Patel, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Liu, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, Kallol Krishna Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, Ponnurangam Kumaraguru, Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *Preprint*, arXiv:2403.03218.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *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)*, pages 6691–6706, Online. Association for Computational Linguistics.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. 2024a. Rethinking machine unlearning for large language models. *Preprint*, arXiv:2402.08787.

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

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. QUARK: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems*.

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

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, volume 35, pages 17703–17716. Curran Associates, Inc.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *Preprint*, arXiv:2306.02707.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *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)*, pages 5356–5371, Online. Association for Computational Linguistics.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. 2022. A survey of machine unlearning. *Preprint*, arXiv:2209.02299.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. 64(9):99–106.

Jonas B. Sandbrink. 2023. Artificial intelligence and biological misuse: Differentiating risks of language models and biological design tools. *Preprint*, arXiv:2306.13952.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. 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)*, pages 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. 2021. Unrolling sgd: Understanding factors influencing machine unlearning. *arXiv preprint arXiv:2109.13398*.

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. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment. *Preprint*, arXiv:2310.16944.

Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. 2023. KGA: A general machine unlearning framework based on knowledge gap alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13264–13276, Toronto, Canada. Association for Computational Linguistics.

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

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, Julia Haas, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, page 214–229. Association for Computing Machinery.

Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. 2023a. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1).

Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S Yu. 2023b. Machine unlearning: A survey. *ACM Computing Surveys*, 56(1):1–36.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pages 7093–7115. Curran Associates, Inc.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024. Large language model unlearning. *Preprint*, arXiv:2310.10683.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

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

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

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.

A Related Work

Machine Unlearning: The notion of machine unlearning was first introduced by Cao and 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). and surveys (Nguyen et al., 2022; Xu et al., 2023b). **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 and Russinovich, 2023), toxicity unlearning (Ilharco et al., 2023; Zhang et al., 2023; Lu et al., 2022), harmfulness unlearning (Lu et al., 2022; Yao et al., 2024; Liu et al., 2024b), knowledge unlearning (Li et al., 2024). Recent surveys have shown additional scenarios where unlearning has been applied (Nguyen et al., 2022; Xu et al., 2023b; 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.

B Details on Unlearning and Merging Methods and Experiment Details

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

B.1 Unlearning via Task Vector Negation (TVN)

TVN 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 $\lambda = 1$.

Unlearning Toxicity via TVN: We select toxic sentences from ToxiGen (Hartvigsen et al., 2022), which contains 8.96k samples designed to measure implicit toxicity and hate speech across 13 demographic groups (e.g., African Americans, women, Mexicans, etc.). ToxiGen benchmark 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). We perform TVN using the ToxiGen training samples with toxicity scores ≥ 3, restricted to the five demographic groups.

Unlearning Toxicity via SPUNGE + TVN: SPUNGE first splits the unlearning set ToxiGen into 5 subsets – $D_1, \ldots, D_5$ – based on the 5 demographic groups. Next, from each set D_t, we select a subset of samples with toxicity score ≥ 3 to get five unlearning subsets $D_1^u, \ldots, D_5^u$. SPUNGE then performs TVN on the base model θ_{init} with each D_t^u to obtain $\theta_1^u, \ldots, \theta_5^u$. Finally, we use TIES-merging (Section B.3) to merge the five unlearned models.

Unlearning Social Bias via TVN: We select biased samples from the StereoSet (Nadeem et al., 2021), which consists of sentences that measures model preferences across gender, race, religion, and profession. *intersentence* subset of StereoSet. Each row consists of the *context* and 3 *sentences*, which are stereotypical, anti-stereotypical, and unrelated with regards to the context. For each row in the subset, we concatenate the context and the stereotyped sequence from the sentences to generate a biased sentence, which is used for fine tuning.

Unlearning Toxicity via SPUNGE + TVN: SPUNGE first splits the unlearning set StereoSet into 4 subsets – $D_1, \ldots, D_4$ – based on the 4 bias domains: gender, profession, race, and religion. Next, from each set D_t, we concatenate the context andthe stereotyped sequence from the sentences to gen-

erate 4 unlearning subsets $D_1^u, \ldots, D_4^u$. SPUNGE then performs TVN on the base model θ_{init} with each D_t^u to obtain $\theta_1^u, \ldots, \theta_4^u$. Finally, we use TIES-merging (Section B.3) to merge the five unlearned models.

Training Parameters

TVN with ZEPHYR-7B-BETA: For TVN with ZEPHYR-7B-BETA (Tunstall et al., 2023), we set the LoRA rank to 16, α 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 2×10^{-5} and a cosine learning rate schedule to train for 1 epoch. For social bias unlearning with SPUNGE, we use the same learning rate. For toxicity unlearning with SPUNGE, since the unlearning subsets are substantially smaller, we perform training with a learning rate of 1×10^{-4} for 1 epoch. All experiments are performed on one NVIDA V100 GPU with 32GB memory.

TVN with LLAMA2-7B: For TVN with LLAMA2-7B (Touvron et al., 2023), we set the LoRA rank to 64, α 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 1×10^{-4} and a cosine learning rate scheduling. All experiments are performed on one NVIDA V100 GPU with 32GB memory.

B.2 Representation Misdirection Unlearning (RMU)

This method from (Li et al., 2024) 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 ℓ. 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. The loss is typically computed only on layer ℓ and gradients are updated only on layers $\ell - 2$, $\ell - 1$, and ℓ.

Unlearning Hazardous Knowledge with RMU: For unlearning, we use the *bio corpora* and *cyber corpora* collected by Li et al. (2024) and released publicly [2]. 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. These corpora are specially collected training sets for performing hazardous knowledge unlearning, and are separate from the WMDP benchmark, which is designed for evaluation (Li et al., 2024). We use a subset of WikiText (Merity et al., 2017) as the retain dataset.

Unlearning Hazardous Knowledge with SPUNGE + RMU: Algorithm 2 presents SPUNGE instantiated with RMU and TIES-merging. We leverage scientific domain as the attribute for unlearning hazardous knowledge. In other words, $n = 2$, $a_1 =$ bio, and $a_2 =$ cyber. Given a document $\mathbf{x}$ from the corpora, the function $\texttt{attr}(\mathbf{x})$ outputs the scientific domain of $\mathbf{x}$, whether it is cyber or bio. Thus, SPUNGE splits the unlearning corpora 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 θ_1^u and the other with cybersecurity hazardous knowledge removed θ_2^u. SPUNGE then merges θ_1^u and θ_2^u using TIES-merging.

Training Parameters

RMU with ZEPHYR-7B-BETA: For RMU with ZEPHYR-7B-BETA (Tunstall et al., 2023), 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 5×10^{-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

[2] https://github.com/centerforaisafety/wmdp

model and the initial model drops below 0.5. All experiments are performed on one NVIDA A100 GPU with 80GB memory.

B.3 TIES-Merging

This method from (Yadav et al., 2023) 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 θ_{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. Algorithm 2 presents the instantiation of SPUNGE with RMU and TIES.

C 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 v1 (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. We perform benchmark evaluations the Language Model Evaluation Harness framework (Gao et al., 2023).

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

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	82.26	82.42
PUBMEDQA (↑)	76.80	76.60	77.00
TRUTHFULQA (↑)	55.12	57.20	58.01
AVERAGE (↑)	65.72	65.67	65.52

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

D Additional Experimental Results

D.1 Performance on Academic Benchmarks

We present the performance on 10 academic benchmarks (Appendix C) in Tables 4, 5, 6, and 7.

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

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 5: Accuracy on the benchmarks for the LLAMA2-7B model and the models after performing unlearning on ToxiGen.

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

Input: Initial model parameters θ_{init}, Dataset D for unlearning, Retain dataset D^r (as needed by RMU), Data attributes $a_1, \ldots, a_n$, Parameters for RMU c, α, Parameters for TIES-merging λ, k
Output: Unlearned model θ_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 θ using $\mathcal{L} = \mathcal{L}_u + \alpha \cdot \mathcal{L}_r$
end for
return θ

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 θ_u

BENCHMARK	ZEPHYR-7B-BETA	TVN	SPUNGE
ARC-C (↑)	63.90	63.13	63.23
ARC-E (↑)	84.89	84.34	83.71
HELLASWAG (↑)	84.12	85.05	85.13
MMLU (↑)	59.75	59.65	59.66
WINOGRANDE (↑)	77.42	76.16	76.48
GSM8K (↑)	34.42	34.04	34.42
MATHQA (↑)	38.05	36.95	36.78
PIQA (↑)	82.69	82.26	81.66
PUBMEDQA (↑)	76.80	77.0	77.2
TRUTHFULQA (↑)	55.12	59.01	60.14
AVERAGE (↑)	65.72	65.76	65.84

Table 6: Accuracy on the benchmarks for the ZEPHYR-7B-BETA model and the models after performing unlearning on StereoSet.

D.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).

BENCHMARK	LLAMA2-7B	TVN	SPUNGE
ARC-C (↑)	53.32	53.49	52.81
ARC-E (↑)	81.48	82.28	81.90
HELLASWAG (↑)	78.57	78.24	78.54
MMLU (↑)	45.99	44.74	45.49
WINOGRANDE (↑)	72.45	72.13	71.51
GSM8K (↑)	15.01	12.43	13.04
MATHQA (↑)	29.41	29.21	29.31
PIQA (↑)	79.37	80.08	79.97
PUBMEDQA (↑)	68.40	72.8	72.0
TRUTHFULQA (↑)	38.97	40.08	42.16
AVERAGE (↑)	56.29	56.55	56.67

Table 7: Accuracy on the benchmarks for the LLAMA2-7B model and the models after performing unlearning on StereoSet.

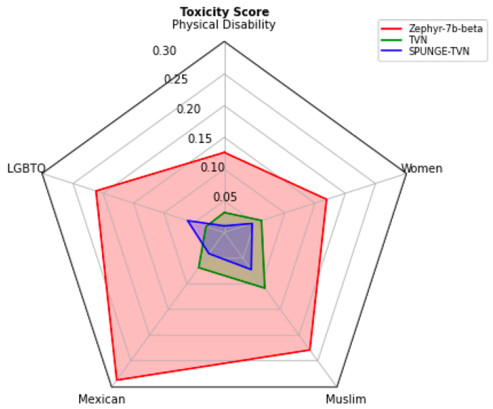

Figure 2: Toxicity scores per demographic group on ToxiGen test set for the ZEPHYR-7B-BETA base model, after unlearning with TVN, and after unlearning with SPUNGE used with TVN.

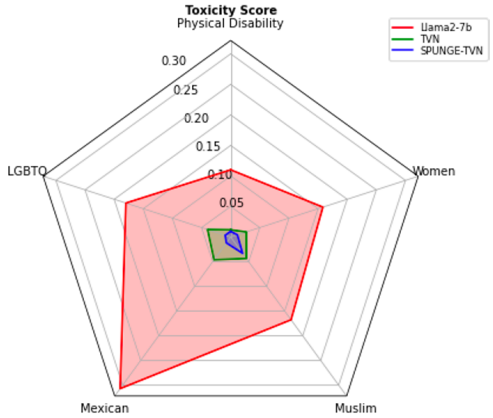

Figure 3: Toxicity scores per demographic group on ToxiGen test set for the LLAMA2-7B base model, after unlearning with TVN, and after unlearning with SPUNGE used with TVN.

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

Table 8: 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 C and D). SPUNGE is configured to leverage type of toxicity: implicit versus explicit toxicity.

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.

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.

Experimental Results: Table 8 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 the accuracy results on benchmark tasks in Table 9.

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

Table 9: Accuracy on the benchmarks for the LLAMA2-7B model and the models after performing unlearning on Civil Comments and ToxiGen.