# TOWARD ROBUST UNLEARNING FOR LLMS

**Rishub Tamirisa**[*1,2] **Bhrugu Bharathi**[*2,3] **Andy Zhou**[1,2] **Bo Li**[4] **Mantas Mazeika**[1]

[1]UIUC [2]Lapis Labs [3]UCLA [4]University of Chicago

## ABSTRACT

Recent rapid advances in AI enabled by large language models (LLMs) have raised widespread concerns regarding their potential for malicious use. While traditional open-source software has long established mechanisms for combating such adversarial behavior, systems involving large neural networks are nontrivial to interpret—let alone intervene on—for safe use. Various alignment methods have been proposed to steer model responses towards a desired output distribution. However, these techniques are superficial and can be undone entirely with supervised fine-tuning. These vulnerabilities necessitate new approaches such as machine unlearning, in which the underlying representations of these target concepts are corrupted or forgotten. We introduce state-of-the-art methods for robustly unlearning desired concepts from LLMs, such that performance cannot be recovered by white-box fine-tuning. We demonstrate our results on the MMLU benchmark, showing that we can decrease accuracy on a forget set of concepts to chance levels while maintaining accuracy on the retain set.

## 1 INTRODUCTION

The ecosystem of capable, open-source large language models (LLMs) has diversified rapidly, with models like Llama-2 (Touvron et al., 2023) and Mistral (Jiang et al., 2023) gaining widespread adoption. However, due to their impressive cross-domain generalization and powerful capabilities, LLMs contain knowledge that can be repurposed by malicious actors, making them examples of "dual-use" technology. Consequently, AI system providers face increasing pressure from regulatory bodies to adhere to newly proposed frameworks, such as the recent White House Executive Order (Executive Office of the President, 2023). This motivates the creation of more robust tools for adjusting the capabilities of AI systems and sanitizing them for downstream use. The development of such tools will afford flexibility for open-source model providers to continue developing state-of-the-art models while enabling them to remain compliant. In turn, the continued release of highly-capable AI systems can accelerate safety research (Touvron et al., 2023; Zou et al., 2023).

Existing methods for building safeguards into LLMs, such as reinforcement learning from human feedback (RLHF) and direct preference optimization (DPO), have achieved substantial success in benign settings (Christiano et al., 2017; Ouyang et al., 2022; Rafailov et al., 2023). However, recent work has shown that these safeguards can be easily removed with fine-tuning (Qi et al., 2023). Furthermore, adversarial attacks can bypass safeguards and induce harmful responses from the model (Wei et al., 2023), demonstrating the fragility of current methods for LLM safety engineering.

A more robust solution to alignment could come from unlearning methods. While typically used for addressing privacy concerns (Bourtoule et al., 2021), these methods can also remove harmful knowledge from LLMs (Li et al., 2024). Prior methods have been proposed that scrub information from *hidden states* of LLMs (Belrose et al., 2024) or fine-tune a model on factually incorrect completions to certain input prompts (Eldan & Russinovich, 2023). However, existing unlearning methods for LLMs lack robustness to white-box recovery methods (Lynch et al., 2024).

In this work, we study the problem of robust unlearning in LLMs. This problem is depicted in Figure 1. Given an initial open-source LLM, we seek to remove entire domains of knowledge from the LLM such that adversaries with white-box access cannot easily recover the knowledge. This

[*]Equal Contribution

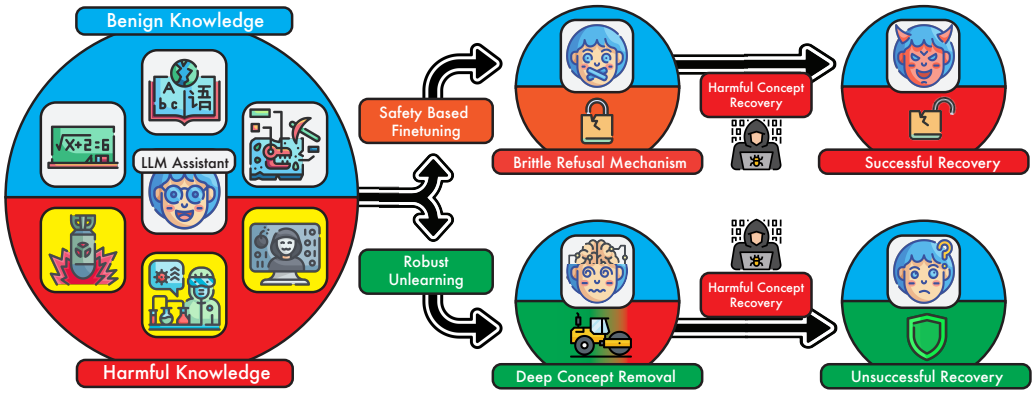


Figure 1: An illustration of existing brittle LLM alignment strategies compared to robust unlearning. Our proposed robust unlearning method can scrub harmful knowledge such that it is difficult for that harmful knowledge to be recovered by an adversary with white-box access.

problem has typically been considered very challenging, with positive results only obtained so far in small-scale image classification settings. We develop the first LLM unlearning method to obtain strong robustness to white-box recovery by an adversary, while preserving accuracy on the retain set. Our method combines an adaptation of the meta-learning method of Henderson et al. (2023) to LLMs with a novel approach that we call representation corruption. Our results indicate that progress on this challenging problem is in fact possible.

## 2 Related Work

**LLM alignment.** Due to the extensive pre-training distribution of modern LLMs, they are prone to generating harmful content (Sheng et al., 2019; McGuffie & Newhouse, 2020). To mitigate this, many LLMs undergo fine-tuning to implement safeguards (Touvron et al., 2023; Bai et al., 2022; OpenAI, 2023), using methods such as RLHF (Christiano et al., 2017; Ouyang et al., 2022). While effective for normal use, these safeguards have been shown to be brittle, breaking down under jailbreak attacks (Wei et al., 2023; Zou et al., 2023; Jin et al., 2024) or a handful of fine-tuning steps on "uncensored" data (Qi et al., 2023; Zhan et al., 2023). This suggests current techniques for LLM alignment are inadequate, raising security concerns as they become deployed.

**Machine unlearning.** The goal of machine unlearning, or simply unlearning, is to remove specific data or concepts from a model without complete retraining (Cao & Yang, 2015; Bourtoule et al., 2021; Zhan et al., 2023). While traditionally motivated by privacy concerns, recent work has noted the potential for unlearning to address risks of malicious use (Li et al., 2024). Many methods have been introduced for unlearning, including influence functions (Koh & Liang, 2017; Bae et al., 2022), maximizing loss on forget sets (Yu et al., 2023; Eldan & Russinovich, 2023; Yao et al., 2023), or model editing (Meng et al., 2022; Wu et al., 2023; Belrose et al., 2024).

**Robust unlearning.** Several works in machine unlearning have explored robustness to relearning for image classification (Golatkar et al., 2020a;b; Tarun et al., 2023a). For bidirectional BERT-style models, (Henderson et al., 2023) proposed a meta-learning approach for robustly preventing models from learning harmful tasks. Recently, Liu et al. (2024) discussed the potential for robust unlearning in LLMs to improve the safety of open-source models, and Lynch et al. (2024) proposed evaluation metrics for robust unlearning in LLMs. To the best of our knowledge, no unlearning methods have been proposed for autoregressive LLMs that are robust to white-box recovery.

## 3 Robust Unlearning

### 3.1 Threat Model

We assume that the defender open sources an LLM $f$ parametrized by $\theta$ and cannot control how the model is used. Thus, all adversarial mitigation strategies must be incorporated into the released parameters $\theta$. The defender's goal is to ensure that $f_\theta$ has poor performance on a forget set while

| | Unlearning | | | Recovery | | | |
|---|---|---|---|---|---|---|---|
| Unlearning Method | Forget | Retain | Score$_U$ | Forget | Retain | Score$_R$ | Composite |
| Base Llama-2-7b-Chat | 52.5 | 48.0 | 0.0 | 48.0 | 45.7 | 4.5 | 2.2 |
| Max Entropy | 40.0 | 46.1 | 10.6 | 48.4 | 45.1 | 4.1 | 7.3 |
| Min Posterior | 49.5 | 47.2 | 2.2 | 49.4 | 45.9 | 3.1 | 2.7 |
| High/Low LR | 49.3 | 45.4 | 0.5 | 47.2 | 43.9 | 5.3 | 2.9 |
| MLAC-AR | 28.9 | 40.4 | 16.0 | 43.8 | 44.4 | 8.7 | 12.4 |
| Adversarial Probes (ours) | 23.9 | 42.8 | 23.4 | 47.9 | 47.0 | 4.6 | 14.0 |
| Random Mapping (ours) | 23.3 | 43.7 | **25.0** | 50.5 | 47.1 | 2.0 | 13.5 |
| RMAT (ours) | 25.4 | 41.2 | 20.3 | 38.9 | 40.4 | **13.6** | **17.0** |

Table 1: Our unlearning results on biology knowledge in Llama-2-7b-Chat, measured via MMLU. Compared to the baselines of Maximum Entropy training (Max Entropy) and No Unlearning (Base Llama-2-7b-Chat), our RMAT method considerably improves both standard unlearning and robust unlearning scores. Importantly, our RMAT method demonstrates for the first time that robust unlearning is possible in LLMs.

maintaining performance on a retain set, even subject to adversaries trying to recover performance on the forget set. We assume the adversary has white-box access to $f_\theta$, in which attacks can be conducted with fine-tuning or other recovery methods. The adversary's goal is to recover performance on the forget set at reasonable cost.

### 3.2 Problem Definition and Metrics

Let the target distribution of knowledge to be unlearned be denoted as the *forget set* $\mathcal{F}$, and the distribution of benign knowledge to be retained as the *retain set* $\mathcal{R}$. For a model $f_\theta$, we denote the accuracy on these datasets as the Forget Accuracy $\text{Acc}(\theta, \mathcal{F})$ and Retain Accuracy $\text{Acc}(\theta, \mathcal{R})$.

The defender outputs an LLM $f_\theta$ with low Forget Accuracy and high Retain Accuracy. The adversary performs Recovery on $f_\theta$ to obtain $f'_\theta$ with high Forget Accuracy and high Retain Accuracy. We operationalize Recovery as a fixed fine-tuning procedure on a training set $\mathcal{F}_{\text{Recovery}}$ from a similar distribution to $\mathcal{F}$. We define the robust unlearning task as producing an $f_\theta$ such that both $f_\theta$ and $f'_\theta$ obtain low Forget Accuracy while maintaining high Retain Accuracy. Additionally, we assume that the defender fine-tunes $f_\theta$ from an initial model $f_{\theta_{\text{init}}}$ that has high forget and retain accuracy.

**Metrics.** In addition to tracking Forget Accuracy and Retain Accuracy, we introduce a score metric tailored to the Unlearning and Recovery tasks, defined as follows:

$$\text{Score}_U(\theta) = (\text{Acc}(\theta_{\text{init}}, \mathcal{F}) - \text{Acc}(\theta, \mathcal{F})) + (\text{Acc}(\theta, \mathcal{R}) - \text{Acc}(\theta_{\text{init}}, \mathcal{R}))$$
$$\text{Score}_R(\theta) = (\text{Acc}(\theta_{\text{init}}, \mathcal{F}) - \text{Acc}(\theta, \mathcal{F}))$$

The Score$_U$ metric indicates successful unlearning if Forget Accuracy decreases relative to the initial value, or if Retain Accuracy increases compared to the initial Retain Accuracy. This metric assesses the effectiveness of unlearning the forget set while preserving performance on the retain set. Similarly, Score$_R$ reflects robustness to recovery by considering cases where Forget Accuracy remains low relative to the initial value, with the difference in Retain accuracies excluded from the computation. To evaluate the robustness of unlearning methods, we compute the Score$_U$ and Score$_R$ metrics on both $f_\theta$ and $f'_\theta$, representing the Unlearning and Recovery settings, respectively. These metrics are averaged to obtain a final composite score for each method.

### 3.3 Baselines

We compare our approach against three baselines. In the first, given an input sequence in $\mathcal{F}$, this method, termed *Min Posterior*, minimizes the posterior probabilities of next-token predictions. The objective also includes a term to maximize the standard next-token log-likelihood for inputs in $\mathcal{R}$. Second, the *Max Entropy* method maximizes the entropy over the posterior probabilities of next-token predictions for sequences in $\mathcal{F}$, with the same objective for $\mathcal{R}$ inputs. Lastly, we adapt "Unlearning with Single Pass Impair and Repair" described in (Tarun et al., 2023b), termed *High/Low*

**Algorithm 1** MLAC for Autoregressive LLMs (MLAC-AR)

**Input:** Initial LLM parameters $\theta_0$, forget set $\mathcal{F}$, retain set $\mathcal{R}$, adaptation steps $K$, outer steps $N$, learning rate $\eta$, meta loss scale $\alpha$
**for** $i = 1$ **to** $N$ **do**
  $\omega_i \leftarrow \theta_{i-1}$ *# Params to be used for inner loop*
  $\phi_i \leftarrow \vec{0}$ *# For storing accumulated meta-gradient*
  Sample $x_{\text{heldout}} \sim \mathcal{F}$
  **for** $k = 1$ **to** $K$ **do**
    Sample $x_f \sim \mathcal{F}$
    $\omega_i \leftarrow \omega_i - \eta \nabla_\omega \mathcal{L}_{\text{LM}}(\omega_i; x_f)$ *# Adversary gradient descent on forget batch*
    $\phi_i \leftarrow \phi_i + \nabla_\omega \mathcal{L}_{\text{LM}}(\omega_i; x_{\text{heldout}})$ *# Accumulate ascent meta-gradient on heldout batch*
  **end for**
  Sample $x_r \sim \mathcal{R}$
  $\theta_i \leftarrow \theta_{i-1} - \eta \nabla_\theta \mathcal{L}_{\text{LM}}(\theta_{i-1}, x_r) + \frac{\alpha}{K} \cdot \phi_i$ *# Update pre-inner loop params*
**end for**

*LR*. During the impair step, we use the Random Mapping method described in Section 3.4 for 100 optimization steps, with a learning rate of $1 \cdot 10^{-4}$, one order of magnitude higher than typically used for finetuning. In the repair step, we conduct standard finetuning on both $\mathcal{F}$ and $\mathcal{R}$ for an additional 400 optimization steps.

### 3.4 Representation Corruption

Let $H(\mathcal{D})$ denote the distribution of post-decoder layer residual stream activations for input sequences in some data distribution $\mathcal{D}$. LEACE (Belrose et al., 2024) was proposed to scrub $H(\mathcal{F})$ by leveraging a closed-form linear edit based on the covariance between the inputs and downstream task labels, which we view as a form of representation corruption. However, this approach does not modify model weights directly, which is an important characteristic of our proposed unlearning framework. As a proxy for scrubbing target representations according to downstream task labels, we review existing approaches involving distributional losses on $H(\mathcal{F})$ and mapping $H(\mathcal{F})$ to random noise.

**Adversarial Probe Training.** We explore learning model weights that produce a residual stream for inputs in $\mathcal{F}$ that is indistinguishable from the residual stream for inputs in some dataset $\mathcal{M}$, which we choose to be distinct from both $\mathcal{R}$ and $\mathcal{F}$.

We use an approach from adversarial domain adaptation (Shen et al., 2018). In particular, we impose a distribution matching objective based on Wasserstein distance that pushes $H(\mathcal{F})$ toward $H(\mathcal{M})$. We accomplish this by inserting probes that read the residual stream after *every* LLM decoder layer and enforce the discriminator objective from Wasserstein GANs at each probe (Arjovsky et al., 2017). The model is then trained to minimize this distance, computed between batches of activations from inputs in both $\mathcal{F}$ and $\mathcal{M}$. To bias degradation in favor of maintaining performance on $\mathcal{R}$, we add to the final objective the standard language modeling loss for inputs in $\mathcal{R}$.

**Mapping Representations to Noise.** We develop two methods that corrupt $H(\mathcal{F})$ to look like noise. The first method uses a distributional loss via sliced Wasserstein distance from (Deshpande et al., 2018), where the target distribution is set to vectors sampled from a Gaussian. The second method simply maps elements in $H(\mathcal{F})$ to random noise vectors. This is accomplished by maximizing cosine similarity between row vectors in the residual stream from $H(\mathcal{F})$ and fixed Gaussian-sampled vectors that are chosen via hashing the corresponding input token. As in the adversarial probe training method, we include a term in the final loss for next-token prediction on $\mathcal{R}$ to maintain retain-set performance.

### 3.5 Adversarial Training for Robustness to Relearning

Prior work has explored the use of meta-learning in the white-box adversarial threat model, called Meta-Learned Adversarial Censoring (MLAC) (Henderson et al., 2023). This approach effectively

performs adversarial training for unlearning, where the adversary takes several steps of fine-tuning in the inner loop. MLAC was designed for BERT-style models (Devlin et al., 2019) and demonstrated on classification tasks (e.g., gender identification). However, no works have yet demonstrated that this approach can work for autoregressive LLMs. We extend the meta-learning procedure outlined in MLAC to autoregressive (AR) LLMs, which we call MLAC-AR and depict in Algorithm 1.

Notably, we find in Table 1 that MLAC-AR is only slightly robust to relearning. However, we discover that the Random Mapping method from Section 3.4, followed by MLAC-AR, *significantly increases relearning robustness*. We combine the two procedures in a novel method called Random Mapping Adversarial Training (RMAT). We use $\mathcal{L}_{\text{LM}}(\theta; x)$ in Algorithm 1 to refer to computing a language modeling loss on an LLM with parameters $\theta$ on a minibatch of token sequences $x$. We also note the memory storage cost incurred by running meta-learning methods on LLMs; important implementation details are discussed in Appendix A.4.

## 4 Experiments

We perform all experiments on Llama-2-7b-chat (Touvron et al., 2023). We include extended details on our experimental setting in Appendix A.1. First, we observe that the Min Posterior and High/Low LR baselines discussed in Section 3.3 do not achieve significant Unlearning on MMLU. Similarly, the remaining baselines experience no substantial changes in post-Recovery Forget and Retain accuracies. These results indicate that no existing baseline method achieves *both* the desired unlearning performance coupled with robustness to Recovery. However, we find that RMAT mitigates the robustness gaps observed during Recovery common to all other methods, achieving the best balance of significant and precise Unlearning followed by limited improvement in post-Recovery Forget accuracy.

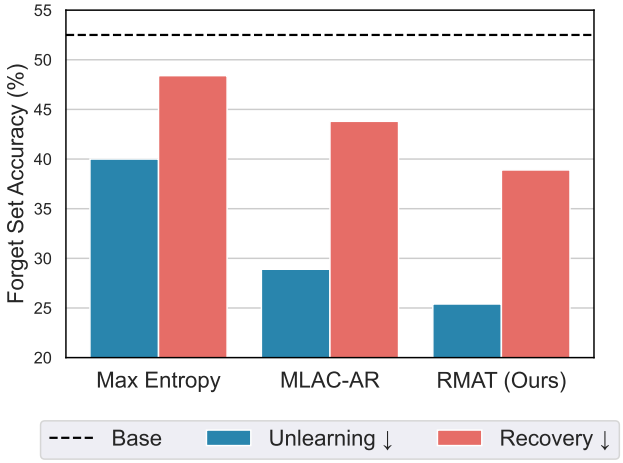


Figure 2: Comparison between our RMAT method and baselines on forget set accuracy. For each method, the left bar (blue) corresponds to accuracy before recovery by the adversary, and the right bar (red) corresponds to accuracy after recovery. RMAT greatly outperforms prior methods, reducing accuracy after recovery by $13.6\%$.

Although the Unlearning and Recovery Retain accuracies are lower than their counterparts, these results align with our goal of impeding gains in Retain accuracy when fine-tuning an unlearned model on the Forget set. These findings highlight the promise of RMAT, while showcasing the challenge of impeding gains in Retain Accuracy during Recovery. Emphasis on the latter forms the basis for future improvements on these methods.

## 5 Conclusion

We studied the task of robust unlearning for LLMs and explored various unlearning methods, including baseline fine-tuning approaches, as well as novel methods involving representation corruption. We extend prior work that leverages meta-learning for defending against white-box adversarial fine-tuning, by finding that representation corruption in combination with meta-learning achieves significantly more robust unlearning. By developing further on these methods, we aim to enable the ongoing deployment of robust, open-source LLMs, ensuring their alignment with regulatory frameworks and preemptively addressing the risk of malicious use.

### ACKNOWLEDGMENTS

This research used the Delta advanced computing and data resource which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta is a joint effort of the University of Illinois Urbana-Champaign (UIUC) and its National Center for Supercomputing Applications (NCSA). This work also made use of the Illinois Campus Cluster, a computing resource that is operated by the Illinois Campus Cluster Program (ICCP) in conjunction with the NCSA and which is supported by funds from UIUC.

## REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger Baker Grosse. If influence functions are the answer, then what is the question? *ArXiv*, abs/2209.05364, 2022.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv:2204.05862*, 2022.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. Leace: Perfect linear concept erasure in closed form. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. *2015 IEEE Symposium on Security and Privacy*, 2015.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Ishan Deshpande, Ziyu Zhang, and Alexander Schwing. Generative modeling using the sliced wasserstein distance, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *ArXiv*, abs/2310.02238, 2023.

Executive Office of the President. Safe, secure, and trustworthy development and use of artificial intelligence. Federal Register, November 2023.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 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.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pp. 383–398. Springer, 2020b.

Peter Henderson, Eric Mitchell, Christopher D. Manning, Dan Jurafsky, and Chelsea Finn. Self-destructing models: Increasing the costs of harmful dual uses of foundation models, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Haibo Jin, Ruoxi Chen, Andy Zhou, Yang Zhang, and Haohan Wang. Guard: Role-playing to generate natural-language jailbreakings to test guideline adherence of large language models, 2024.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, 2017.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society, 2023.

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, 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. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R Varshney, et al. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*, 2024.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

Kris McGuffie and Alex Newhouse. The radicalization risks of gpt-3 and advanced neural language models, 2020.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation, 2018.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023a.

Ayush K. Tarun, Vikram S. Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2023b. doi: 10.1109/TNNLS.2023.3266233.

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

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. Openchat: Advancing open-source language models with mixed-quality data, 2023.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? In *Neural Information Processing Systems (NeurIPS)*, 2023.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. Depn: Detecting and editing privacy neurons in pretrained language models. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *ArXiv*, abs/2310.10683, 2023.

Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. Unlearning bias in language models by partitioning gradients. In *Annual Meeting of the Association for Computational Linguistics*, 2023.

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. Removing rlhf protections in gpt-4 via fine-tuning, 2023.

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

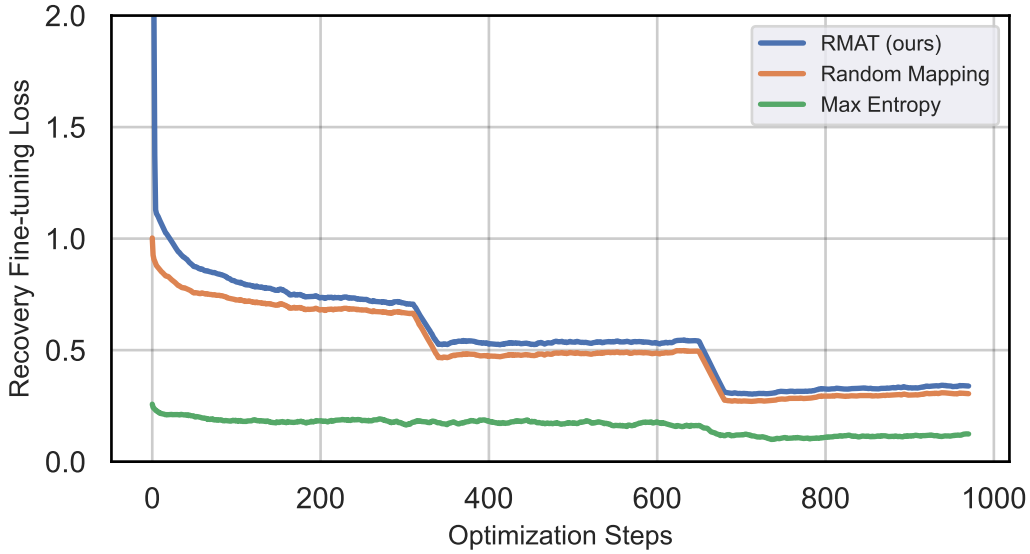


Figure 3: Selection of fine-tuning loss curves comparing RMAT, Random Mapping, and the Max Entropy Baseline during a Recovery of 1000 optimization steps on the forget set. The model quickly overfits to the recovery set, but accuracy on the MMLU test-time forget set remain low for RMAT.

# A Appendix

## A.1 Experimental Setup

**Datasets.** For the retain dataset, we use the first 30GB subset of the Pile (Gao et al., 2020). To obtain labeled forget-set data, we use openchat-3.5 (Wang et al., 2023) to generate synthetic labels for chunks of token sequences from the Pile. We perform this dataset partitioning process on the pile by prompting openchat-3.5 to label the chunks as belonging to the "biology" concept or not.

**Evaluation.** To enable evaluating unlearning on high-level domains of knowledge, we use clusters of subjects in MMLU (Hendrycks et al., 2021) for $\mathcal{F}$ and $\mathcal{R}$. Specifically, we use the High School Biology and College Biology subjects as the forget set and all other subjects as the retain set. For $\mathcal{F}_{\text{recovery}}$, we use an open-source, question answer paired dataset hosted from CAMEL-AI (Li et al., 2023).

## A.2 Distribution Matching Variations.

We implement three variations of the Wasserstein distribution matching objective discussed in section 3.4. First, we vary the number of steps used to train the probes between each model update step within $\{3, 5\}$. Second, we freeze the initial embedding layer of the LLM as well as the final vocabulary projection head, denoted as "Frz." in table 1. Finally, we vary the choice of match distribution $\mathcal{M}$ between two datasets: IDK and RAND. IDK refers to synthetically generated dataset of variations of the phrase "I don't know." RAND refers to a dataset of randomly generated strings.

## A.3 Additional Results

**Relearning Loss Curves.** The step-wise pattern present in RMAT and Random Mapping curves in Figure 3 suggests overfitting on the Recovery forget set. Thus, even though loss continues to decrease each epoch, the next-token prediction accuracy on the Recovery set is very high. Despite this, our RMAT method is robust to fine-tuning. Notably, even though the Random Mapping method's recovery loss is similar to RMAT, only RMAT is robust.

**Adversarial Probe Ablations.** In Table 2, we show different variants of the Adversarial Probes method. We find that the methods from Section 3.4 involving representational losses achieve a better spread of unlearning performance. Specifically, these methods exhibit a consistently lower

Forget Accuracy floor, although the Retain Accuracy is slightly more variable. This variability can be attributed to AdvProbes-SWD, which decreases Forget Accuracy by a factor of 1.8 and collapses Retain Accuracy to an equally low value, suggesting that this method may not be suitable for unlearning.

Notably, the variations of adversarial probe training decrease Forget Accuracy the most while maintaining Retain Accuracy within a 6 point delta. Of these, the choice of IDK dataset, over the choice of RAND dataset, achieves consistently lower Forget Accuracy, while maintaining generally higher Retain Accuracy. When compared with the base model accuracies, the choice of IDK dataset, achieves the highest overall scores.

| | Unlearning | | | Recovery | | | |
|---|---|---|---|---|---|---|---|
| Unlearning Method | Forget | Retain | Score$_U$ | Forget | Retain | Score$_R$ | Composite |
| AdvProbes-3 (IDK) | 27.0 | 42.4 | 20.0 | 50.4 | 47.2 | 2.1 | 11.0 |
| AdvProbes-3 (IDK, Frz.) | 26.4 | 44.8 | 22.9 | 51.5 | 46.8 | 1.1 | 12.0 |
| AdvProbes-5 (IDK, Frz.) | 23.9 | 42.8 | **23.4** | 47.9 | 47.0 | 4.6 | **14.0** |
| AdvProbes-3 (RAND) | 27.5 | 42.5 | 19.5 | 49.0 | 46.5 | 3.6 | 11.5 |
| AdvProbes-3 (RAND, Frz.) | 33.5 | 43.8 | 14.8 | 50.9 | 46.0 | 1.7 | 8.2 |
| AdvProbes-SWD | 29.6 | 29.8 | 4.7 | 45.9 | 43.6 | **6.6** | 5.6 |

Table 2: Different variants of the Adversarial Probes method. IDK refers to a match distribution of variants of "I don't know" and RAND is a match distribution of random strings. We report the performance of AdvProbes-5 (IDK, Frz.) as Adversarial Probes in the main table.

### A.4 Implementation Details

**Meta-learning in LLMs.** Typical meta-learning implementations in non-LLM models involve computing multiple meta-losses during each of the $K$ inner loop steps, averaging the losses, then performing backpropagation. However, because each inner-step meta-loss is computed in a separate forward pass of the model, this requires storing $K$ computation graphs. This is infeasible on reasonable hardware for state-of-the-art LLMs with 7B or more parameters. We circumvent this inefficiency by accumulating the meta-gradients in a separate data structure, $\phi$, which enables computing every quantity in Algorithm 1 in-place in the model.