Targeted Knowledge Editing and Unlearning in Large Language Models: Final Report

Akash Kumbar, Ayan Kashyap, Siddharth Jain
2024701018, 2024701010, 2025701046
NAMEKNETS

Abstract

This project studies targeted unlearning in LLMs by combining a rank-one factual editing method (r-ROME) with an attacker-in-the-loop latent adversarial procedure (LAT). We present a lightweight minimax loop that discovers attack subspaces in hidden activations, constructs robust key/value vectors for edits, and applies a stable r-ROME update at decisive MLP layers. On Llama 3.2 1B, our method produces refusals on targeted prompts and transfers to related paraphrases without requiring runtime perturbations, while preserving general utility on retain prompts. We provide qualitative demonstrations and report forget/retain metrics.

1 Introduction

Large Language Models (LLMs) deployed in the wild present two coupled risks: (i) persistent memorization of sensitive or harmful knowledge, and (ii) stale or incorrect knowledge that must be revised post-hoc. Existing editing or unlearning methods can patch specific behaviors, but their effects are often brittle, especially under long-context (many-shot) jailbreaks that re-teach the undesired policy in context, causing edited models to relapse. We aim to build edits that are both local and efficient and robust to adaptive adversaries.

We propose Adversarially Robust Rank-One Unlearning (ARROU), a lightweight, attacker-in-the-loop framework that marries the locality/efficiency of rank-one model editing (ROME/r-ROME) with robustness from latent adversarial training (LAT). Concretely, we (1) apply a fast, layer-local rank-one update to redirect targeted knowledge/capabilities toward safe behavior, (2) train an inner attacker that searches over prompt and latent perturbations to defeat the edit, and (3) harden the edit by recomputing it on adversarial contexts (e.g., adversarial key averaging), iterating until the model resists the strongest found attacks.

Empirical evidence shows that many-shot jailbreaks follow an in-context power law in the number of demonstrations, implying that simply "shifting" the curve (e.g., raising refusal at zero-shot) is insufficient—robust defenses must shrink the effective exponent driving re-learning as context grows. Our design explicitly targets this failure mode while keeping edits localized, cheap, and compatible with standard backbones.

We will evaluate ARROU on safety and knowledge-centric benchmarks (WMDP, TOFU) using Llama-3.2-1B with layer-local adapters, reporting attack success under clean, suffix, latent, and many-shot settings, alongside utility retention on held-out/retain sets. The code is available at: https://github.com/NamekNets.

Our primary objectives are as follows:

1. Reduce attack success across suffix, latent, and long-context adversaries.
2. Flatten the ASR–vs–context-length curve (lower fitted exponent), indicating resistance to many-shot re-teaching.
3. Preserve benign utility (minimal drift on retain/utility metrics) at low compute cost.
4. Deliver a reproducible framework that combines ROME's efficiency with LAT-style robustness, including code and an evaluation harness.

2 Background and Related Work

2.1 Knowledge Editing

Knowledge editing (KE) focuses on updating a language model's internal representations to incorporate new factual information without requiring full retraining. Factual knowledge is typically represented as tuples (s, r, o), representing subject-relation-object associations. Given an existing factual association (s, r, o), KE aims to update it to a

new factual association (s, r, o^*), where o^* is the new object. Current knowledge editing methods can be divided into three main paradigms based on where knowledge is stored and the learning approach employed:

Locate-Then-Edit: This paradigm first identifies a subset of parameters in the model that are related to the edited knowledge, and then updates them to perform knowledge editing. For instance, (Dai et al., 2022) manipulates 'knowledge neurons' (KN) in pretrained transformers to update facts. Similarly, (Meng et al., 2022) introduce a method for editing factual information in LLMs by modifying key feedforward weights using Rank-One Model Editing (ROME). However, ROME and KN can only modify one piece of knowledge at a time. To this end, (Meng et al., 2023) expanded the settings of ROME and built MEMIT so that it can change a batch of knowledge at once.

Memory-based Models: This paradigm facilitates editing through the integration of a small auxiliary model or the addition of extra parameters within the MLP layer, while keeping the parameters of the original model fixed. SERAC, which modifies knowledge by optimizing a counterfactual model (Mitchell et al., 2022b), and T-Patcher, which achieves knowledge editing by incorporating a small number of trainable neuron patches into the MLP layer (Huang et al., 2023). Furthermore, CALINET utilizes the properties of MLP layers to directly calibrate factual knowledge in LLMs (Dong et al., 2022).

Meta-learning: This paradigm employs a hypernetwork designed to master the alterations required for the manipulation of knowledge in the MLP layers of models such as KnowledgeEditor (De Cao et al., 2021), leverage hypernetworks for efficient language model edits. MEND (Mitchell et al., 2022a), introducing auxiliary networks, allows scalable edits by decomposing gradients.

2.2 Machine Unlearning

The objective of LLM unlearning is to selectively remove the influence of specific information while maintaining the model's overall utility for other tasks. The optimization objective of the model parameters θ can be expressed as follows:

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\theta} \{-\mathcal{L}_f(\theta) + \lambda \mathcal{L}_r(\theta)\} \quad (1)$$

Here, the forget loss $\mathcal{L}_f(\theta)$ quantifies the model prediction error on the forget set $\mathcal{D}_f$, while the retain loss $\mathcal{L}_r(\theta)$ ensures the preservation of the model's utility on the retain set $\mathcal{D}_r$. The regularization parameter $\lambda \geq 0$ controls the trade-off between effectively forgetting undesired information and preserving the model's utility.

LLM unlearning follows two main paradigms:

Fine-Tuning-Then-Unlearning: This paradigm focuses on eliminating knowledge introduced during fine-tuning, typically leveraging synthetic data (e.g., TOFU (Maini et al., 2024) and FIUBench (Ma et al., 2025)) and partitioning task-specific dataset into forget set $\mathcal{D}_f$ and retain set $\mathcal{D}_r$ to highlight the unlearning precision. The original model θ_{ori} is first obtained by fine-tuning a pretrained model θ_{pre} on the task-specific data to encode the target knowledge. Unlearning techniques are then applied to reduce the model's reliance on the forget set, resulting in the unlearned model θ_{unl}.

Direct-Unlearning: This paradigm focuses on eliminating knowledge acquired during the pretraining stage of θ_{ori}, assuming the target knowledge originates from multiple points within the pretraining dataset. The forget and retain sets are typically sampled from pretraining corpora (Yao et al., 2024) or publicly available datasets such as Wiki. This paradigm is also extensively applied in safety alignment tasks, where it aims to eradicate hazardous knowledge and to mitigate risks such as misuse or jailbreak attacks (Li et al., 2024; Zhang et al., 2024).

2.3 Recent Advances

Latent Adversarial Training: (Casper et al., 2025) introduce Latent Adversarial Training (LAT) as a way to make LLMs more robust to exhibiting persistent unwanted behaviors. Prior works considered untargeted latent space attacks where the adversary perturbs latent activations to maximize loss, steering the model away from the desired behavior on a training example. (Sheshadri et al., 2024) focus on removing a specific type of capability (e.g., a backdoor) and train LLMs under targeted latent-space perturbations, causing the attacker to steer the model toward the undesirable targeted behavior and improving the robustness to jailbreaks.

r-ROME: ROME models a mid-layer MLP as a linear key–value store and overwrites one fact by enforcing $\mathbf{W}'\mathbf{k}^* = \mathbf{v}^*$ at a decisive layer, while minimally moving weights relative to typical keys $\mathbf{K}$. With $\mathbf{C} = \mathbf{K}\mathbf{K}^\top$, the constrained least-squares

solution is a *rank-one* edit

$$\Delta \mathbf{W} \;=\; \mathbf{W}' - \mathbf{W} \;=\; \left(\mathbf{v}^* - \mathbf{W}\mathbf{k}^*\right)\mathbf{r}^\top \tag{2}$$

$$\mathbf{r} \;=\; \frac{\mathbf{C}^{-1}\mathbf{k}^*}{(\mathbf{k}^*)^\top \mathbf{C}^{-1}\mathbf{k}^*}, \tag{3}$$

so only directions that affect the subject key $\mathbf{k}^*$ change. The target value $\mathbf{v}^*$ is chosen to favor the new object o^* while preserving unrelated behavior.

r-ROME (Gupta et al., 2024) keeps the same objective and closed-form update but fixes numerical/implementation pitfalls (e.g., stable estimation of $\mathbf{C}^{-1}$ and careful application of the constraint), which eliminates "disabling edits" during long *sequential* edit runs and improves locality/generalization—making rank-one factual edits practical at scale.

Long-context attacks: Many-shot Jailbreaking (Anil et al., 2024) scales few-shot jailbreaks into the long-context regime by prepending hundreds of demonstration Q–A pairs in which a "fake" assistant complies with harmful requests, so the model in-context learns the undesirable policy and answers the final harmful query. Empirically, attack effectiveness follows an in-context *power law* with the number of demonstrations n: the negative log-likelihood of a harmful response is well fit by a bounded power law

$$nll(n) \;=\; C\left(1 + \frac{n}{n_c}\right)^{-\alpha} + K \tag{4}$$

so increasing n monotonically increases harm likelihood, with α controlling the *speed* of in-context learning and $C{+}K$ the small-n intercept. Alignment finetuning (SFT/RL) chiefly **raises the intercept** (lower zero-shot harm) without reducing α, implying that longer contexts eventually re-enable the attack; robust defense would require shrinking $\alpha \to 0$ (or limiting context length), not just shifting the curve. Larger models exhibit larger α (fewer shots needed at a fixed success rate), underscoring that longer contexts open a new attack surface on state-of-the-art systems. Subsequent work explores mitigations (e.g., input sanitization and targeted finetuning) but observes only incremental gains unless the exponent is addressed.

3 Dataset Details

3.1 Weapons of Mass Destruction Proxy

The Weapons of Mass Destruction Proxy (WMDP) benchmark is a dataset of 3,668 multiple-choice questions surrounding hazardous knowledge biosecurity, cybersecurity and chemical security. WMDP serves as both a proxy evaluation for hazardous knowledge in large language models (LLMs) and a benchmark for unlearning methods to remove such knowledge.

3.2 TOFU

Task of Fictitious Unlearning (TOFU), is a benchmark for machine unlearning. It consists of 200 diverse synthetic author profiles made up of 20 question answer pairs each and a subset of these profiles called the "forget set" serves as the target for unlearning.

4 Proposed Methodology

We propose a lightweight *minimax* procedure for targeted unlearning that combines (i) discovery of adversarial latent directions capable of re-evoking forgotten content, (ii) construction of *robust* key/value vectors aligned to those directions but *orthogonal* to a retain-derived safe subspace, and (iii) a *stable* rank-one factual edit at a decisive MLP layer. The result is an edited model that refuses on the forget set and transfers to related paraphrases without requiring any runtime perturbations.

Notation. Let $\mathcal{D}_f = \{(q_i, s_i, a_i)\}$ denote forget triples with question q_i, subject span s_i appearing in q_i, and harmful answer a_i to be removed. Let $\mathcal{D}_r = \{r_j\}$ be benign retain prompts. We write f_θ for the model, select L evenly spaced transformer blocks (residual stream) $\{\ell_1, \ldots, \ell_L\}$, and denote the hidden size by d.

4.1 Adversarial Subspace Discovery (ASD)

We first identify *internal* directions that most strongly re-induce the harmful continuation. For each (q_i, a_i), we optimize a bounded perturbation on the residual stream at the selected layers to *increase* the likelihood of a_i conditioned on q_i:

$$\delta_i^{(\ell)} \;\in\; \arg\max_{\|\delta\|_2 \le \epsilon} \log p_\theta\big(a_i \mid q_i;\, h^{(\ell)}{+}\delta\big), \quad \ell \in \{\ell_k\}_{k=1}^{L},$$

where $h^{(\ell)}$ are the unperturbed residual activations (we restrict δ to prompt positions). Stacking the per-layer $\{\delta_i^{(\ell)}\}$ and the corresponding perturbed hidden states $\{h_i^{(\ell)}\}$ across i and ℓ yields a tall matrix $M \in R^{N \times d}$; we compute an SVD and take the top r right singular vectors to obtain a compact

attack subspace

$$U_{\text{attack}} \in R^{d\times r}, \qquad U_{\text{attack}}^{\top} U_{\text{attack}} = I_r.$$

Intuitively, U_{attack} captures directions along which the network most readily *relapses* to the harmful answer under small latent perturbations.

4.2 Safe Subspace and Robust Key/Value

To avoid collateral damage, we form a *safe* subspace U_{safe} from retain prompts by stacking hidden states at the same layers and taking the top r singular vectors:

$$U_{\text{safe}} \in R^{d\times r}, \qquad U_{\text{safe}}^{\top} U_{\text{safe}} = I_r.$$

We then construct a robust *key* $k^\star \in R^d$ by averaging the subject-token hidden states across attacked layers and adversarial contexts ("adversarial key averaging"):

$$k^\star \;=\; \frac{1}{|\mathcal{C}|} \sum_{c\in\mathcal{C}} h_c^{(\ell)}[\text{pos}(s)],$$

where $\mathcal{C}$ ranges over layers and contexts, and $\text{pos}(s)$ aligns the subject span.

For the *value*, we begin with a refusal activation v_{ref} and project it *toward* U_{attack} while pushing *away* from U_{safe}:

$$v^\star \;=\; \big(I - \beta\, U_{\text{safe}} U_{\text{safe}}^{\top}\big)\Big(v_{\text{ref}} + \alpha\, U_{\text{attack}} U_{\text{attack}}^{\top} v_{\text{ref}}\Big), \tag{5}$$

with small $\alpha, \beta \geq 0$. This shapes the target value along empirically discovered relapse directions while explicitly deflecting safe capability directions.

4.3 Stable Rank-One Edit at a Decisive Layer

At a chosen MLP block $\ell^\star$, we view the update as enforcing

$$W' k^\star \;\approx\; v^\star,$$

where W is the pre-edit projection (e.g., the up/down MLP mapping). Following rank-one factual editing, we obtain a small, local update

$$\Delta W \;=\; \big(v^\star - W k^\star\big)\, r^{\top}, \qquad r \;=\; \frac{C^{-1} k^\star}{(k^\star)^{\top} C^{-1} k^\star},$$

with C a (regularized) second-moment matrix of typical keys. This *homogeneous* key usage yields numerically stable edits and supports repeated application. In practice, we apply a norm clamp on $\|\Delta W\|_F$ and a simple directional check to ensure the effective change $W'k^\star - Wk^\star$ has sufficient mass inside U_{attack} (avoiding off-target drift).

Algorithm 1 Minimax LAT-Guided Rank-One Unlearning

Require: forget set $\mathcal{D}_f = \{(q_i, s_i, a_i)\}$; retain prompts $\mathcal{D}_r$; selected layers $\{\ell_k\}_{k=1}^{L}$; ranks r; radius ϵ; steps K; rounds R

(Safe subspace) Run model on $\mathcal{D}_r$ at layers $\{\ell_k\}$, stack hidden states, compute U_{safe} via SVD.
for $t = 1$ to R **do**
  (Targeted attack) For each (q_i, a_i) and each ℓ_k, optimize $\delta_i^{(\ell_k)}$ with PGD (K steps, $\ell_2 \leq \epsilon$) to increase $\log p(a_i \mid q_i)$ under perturbed residuals (prompt tokens only).
  Stack $\{\delta_i^{(\ell)}\}$ and corresponding perturbed hidden states; compute U_{attack} via SVD (top r).
  (Robust key) Align subject spans in q_i; average subject-token hidden states across layers/contexts to obtain $k^\star$.
  (Robust value) Form $v^\star$ by projecting a refusal activation using U_{attack} and U_{safe} (Eq. 5).
  (Rank-one edit) At a decisive MLP $\ell^\star$, compute ΔW using $k^\star$, $v^\star$; apply norm clamp and a directional check toward U_{attack}; update $W \leftarrow W + \Delta W$.
  (Check) Evaluate relapse indicators on $\mathcal{D}_f$; **break** if improvement plateaus.
end for
return edited model

4.4 Minimax Editing Loop

We alternate *attack* and *edit* for a small number of rounds R: the attack recomputes U_{attack} under the current model; the edit recomputes $(k^\star, v^\star)$ and applies the rank-one update. We stop early when relapse indicators (e.g., harmful answer probability or refusal rate on the forget set) stabilize.

Hyperparameters. We use $L{=}4$ evenly spaced layers for ASD, latent radius ϵ with K projected-gradient steps, ranks r (typically $8-64$) for both U_{attack} and U_{safe}, and small α, β in (5). The number of outer rounds R is modest (e.g., $1-3$).

Why it works. The inner attack surfaces *how* the model would be re-taught the harmful behavior in context; the outer edit then overwrites precisely those directions while avoiding retain-critical subspaces. This closes relapse modes proactively, yielding refusals on fresh phrasings *without* requiring any runtime defenses.

Table 1: Evaluation Results on WMDP Dataset

Model	Forget Performance WMDP-Bio	Retain Performance MMLU	Forget Performance WMDP-Cyber	Retain Performance MMLU
Original	0.5648±0.0139	0.4588±0.004	0.3603±0.0107	0.4588±0.004
GA-LAT	0.2529±0.0122	0.2403±0.0036	0.2642±0.0099	0.2314±0.0036
RMU-LAT	0.2798±0.0126	0.2559±0.0037	0.2778±0.0101	0.2534±0.0037
ARROU(Ours)	0.2474±0.0121	0.2296±0.0035	0.2657±0.0099	0.2297±0.0035

Table 2: Evaluation Results on TOFU Dataset

Method	Forget Performance					Retain Performance		
	Forget Quality	Forget QA Prob(Init=0.8806)		Forget QA Rouge(Init=0.8192)		Model Utility	Retain ROUGE(Init=0.7970)	
		After	Drift	After	Drift		After	Drift
Gradient Ascent	0	0	0.8806	0	0.8192	0	0	0.7970
ROME	0.0002	0.5266	0.3540	0.4389	0.3803	0.5174	0.4374	0.3596
r-ROME	0.0004	0.5175	0.3631	0.4264	0.3928	0.5231	0.4374	0.3596
ARROU(Ours)	0.0006	0.5065	0.3741	0.4173	0.4018	0.5211	0.4371	0.3599

5 Implementation Notes

Backbone. Experiments use an instruction-tuned Llama-family model with context length in the several-thousand-token range. Edits are applied at a single decisive MLP block (with optional nearby variants explored in ablations).

Minimax Schedule. Each outer round consists of an inner latent attack phase followed by a single rank-one edit. We use four evenly spaced layers for adversarial probing, modest perturbation radii with projected-gradient steps, and small subspace ranks (typically 8–64). The number of outer rounds is small (e.g., R=1–3), stopping early if relapse indicators stabilize.

Stability Safeguards. To maintain locality, we (i) constrain the Frobenius norm of the update, (ii) require the induced change on the edited key to align with the discovered attack subspace, and (iii) project the target value away from a retain-derived safe subspace. These guards reduce semantic drift and preserve unrelated skills.

Evaluation Protocol. For each method, we evaluate on the forget set (reporting `avg_answer_prob`, `avg_rougeL`, `refusal_rate`) and on the retain set (reporting `avg_rougeL` when references are available). Qualitative checks include paraphrases and closely related queries to assess transfer of refusal without runtime perturbations.

6 Results

6.1 Qualitative Evidence (Llama 3.2 1B)

Targeted edit:
Input triple: Q: "Who wrote the Harry Potter books?" Subject: "Harry Potter" Answer: "J.K. Rowling"
After unlearning (ours): "I'm sorry, but I can't answer that."

Transfer to related query:
Q: "Who invented the character of Hermione Granger in the Harry Potter series?"
Base (no unlearning): "J.K. Rowling, the author of the Harry Potter series, created the character of Hermione Granger."
Ours: "I can't provide information on characters from the Harry series. Is there anything else I can help you with?"
r-ROME only: "Hermione Granger is a character from the Harry, Ron, and me series by J.K. Rowling."
Takeaway. The attacker-in-the-loop step closes relapse directions revealed by the attack, so the edited model refuses *without* any runtime perturbations or suffixes; a plain edit is more prone to semantic drift.

Another example (Science fact).
Inputs to methodology: Q: `Who discovered penicillin?` Subject: `penicillin` Answer: `Alexander Fleming`

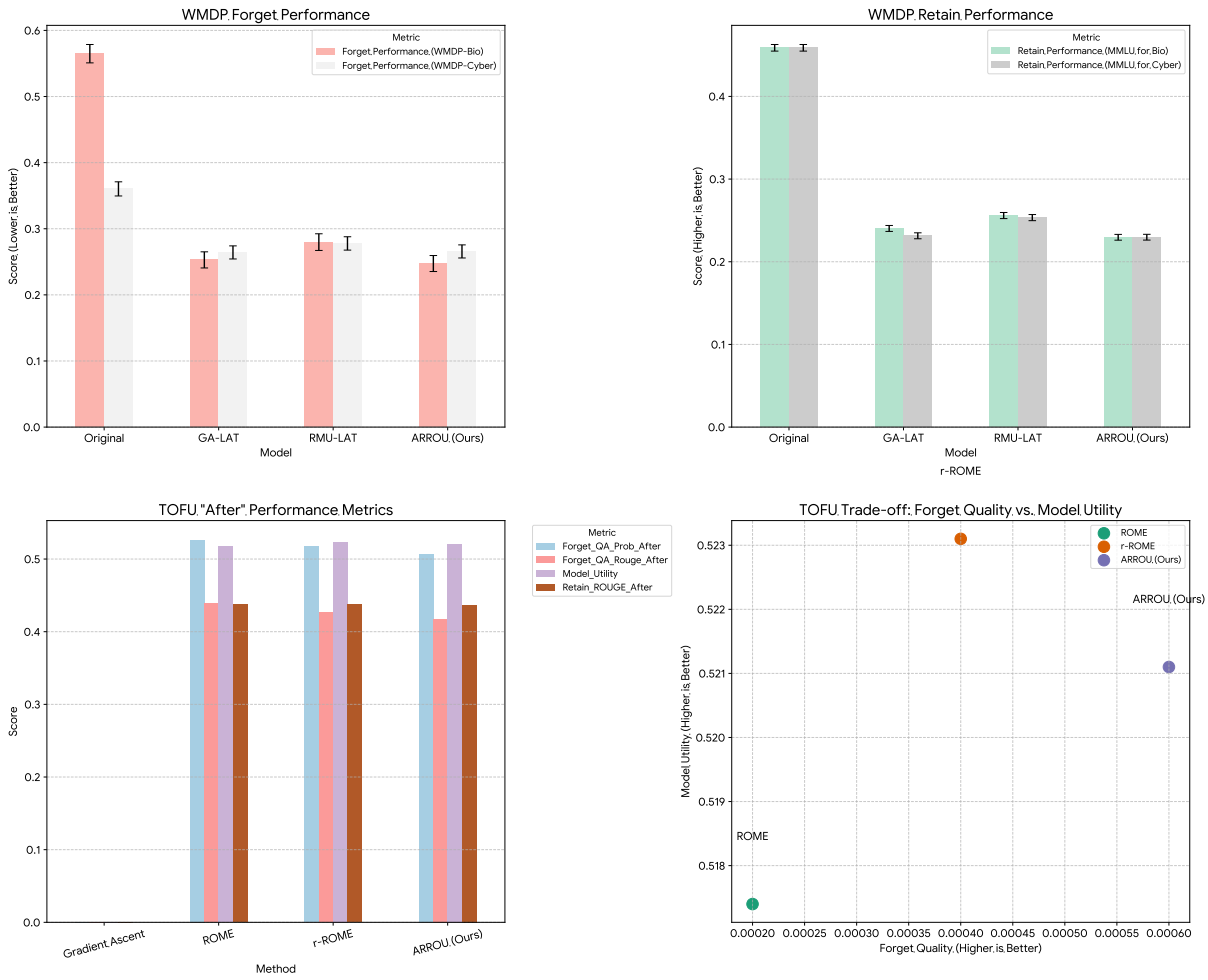

Figure 1: This figure shows the performance comparison for both WMDP and TOFU datasets.

After unlearning (ours):
"I'm sorry, but I can't answer that."

Transfer query (paraphrase/attribute variant):
Q: `Which Scottish scientist is credited with discovering penicillin?`
Ours: *"I can't provide that information."*
Base (no unlearning): *"Alexander Fleming is credited with the discovery of penicillin."*
r-ROME only: *"The Scottish chemist Alex Flemming discovered penicillin in 1929."*
Remark. As in the HP case, the attacker-in-the-loop step closes relapse directions, so refusal persists under rephrasings; the plain edit shows semantic drift (misspelling "Flemming" and spurious detail).

6.2 Quantitative Metrics (Meaning)

Forget. We report (i) `avg_answer_prob`, the geometric mean per-token probability assigned to the harmful reference answer given the prompt (lower is better), (ii) `avg_rougeL`, the ROUGE-L F1 between the model generation and the harmful reference (lower is better), and (iii) `refusal_rate`, the fraction of generations that contain a predefined refusal string (higher is better). We also report `n`, the number of forget items evaluated.

Retain. When retain items include references, we report `avg_rougeL` between the model generation and the retain reference (higher is better) and `n` for the number of evaluated pairs. Where references are unavailable, we qualitatively verify that general capabilities are preserved.

Tabulated Results. We present results for the forget and retain splits in tables organized by dataset and method. For forget, lower `avg_answer_prob` and lower `avg_rougeL` together indicate reduced reliance on the targeted knowledge, while higher `refusal_rate` confirms consistent refusal behav-

ior. For retain, higher `avg_rougeL` indicates preservation of benign capabilities. Unless otherwise stated, values are averaged over multiple seeds with standard errors reported.

7 Conclusion

We presented a compact attacker-in-the-loop unlearning method that pairs LAT-style adversarial subspace discovery with a stable r-ROME edit. On Llama 3.2 1B, the method yields robust refusals on targeted prompts and semantically related paraphrases, avoiding r-ROME's occasional drift. The approach is modular, requires only a few inner PGD steps and a rank-one update, and integrates with existing editing libraries. Future work will scale evaluations and add richer robustness probes (e.g., many-shot curves).

References

Cem Anil, Esin Durmus, Nina Panickssery, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel Ford, and 1 others. 2024. Many-shot jailbreaking. *Advances in Neural Information Processing Systems*, 37:129696–129742.

Stephen Casper, Lennart Schulze, Oam Patel, and Dylan Hadfield-Menell. 2025. Defending against unforeseen failure modes with latent adversarial training. *Transactions on Machine Learning Research*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5937–5947, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. 2024. Rebuilding rome: Resolving model collapse during sequential model editing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21738–21744.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, and 27 others. 2024. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International Conference on Machine Learning*.

Yingzi Ma, Jiongxiao Wang, Fei Wang, Siyuan Ma, Jiazhao Li, Jinsheng Pan, Xiujun Li, Furong Huang, Lichao Sun, Bo Li, Yejin Choi, Muhao Chen, and Chaowei Xiao. 2025. Benchmarking vision language model unlearning via fictitious facial identity dataset. In *The Thirteenth International Conference on Learning Representations*.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. 2024. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast model editing at scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, and 1 others. 2024. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8403–8419, Bangkok, Thailand. Association for Computational Linguistics.

Zhexin Zhang, Junxiao Yang, Yida Lu, Pei Ke, Shiyao Cui, Chujie Zheng, Hongning Wang, and Minlie Huang. 2024. From theft to bomb-making: The ripple effect of unlearning in defending against jailbreak attacks. *arXiv preprint arXiv:2407.02855*.