Collapse of Irrelevant Representations (CIR) Ensures Robust and Non-Disruptive LLM Unlearning

Filip Sondej*
Jagiellonian University

Yushi Yang
University of Oxford

Abstract

Current unlearning techniques and safety training consistently fail to remove dangerous knowledge from language models. We analyze the root causes and propose a highly selective technique which unlearns robustly and without disrupting general performance.

We perform PCA on activations and output gradients to identify subspaces containing common representations, and *collapse them before calculating unlearning updates*. This way we avoid unlearning general representations, and only target those specific to the unlearned facts.

When unlearning WMDP dataset facts from Llama-3.1-8B, we drop post-attack accuracy 30x more than SOTA (Circuit Breakers) on biohazardous facts and 6x more on cyberhazardous facts. Despite this, we *disrupt general performance 30x less*, while requiring less than 3 GPU-seconds per fact.

Code: github.com/filyp/unlearning

1 Introduction

During pre-training, language models learn hazardous capabilities useful e.g. for bioterrorism and cybercrime [Li et al., 2024]. They even acquire information about their own safety controls, which in the future could let models circumvent them [Roger, 2024, Greenblatt et al., 2024].

Popular safety training approaches like DPO and RLHF do not eliminate unwanted capabilities, but rather teach the model to stop using them Lee et al. [2024]. These concealed capabilities can be resurfaced by jailbreak attacks [Zou et al., 2023] or even completely accidentally [Qi et al., 2023]. Even methods designed specifically for unlearning can be easily reversed [Łucki et al., 2025, Lynch et al., 2024, Deeb and Roger, 2024].

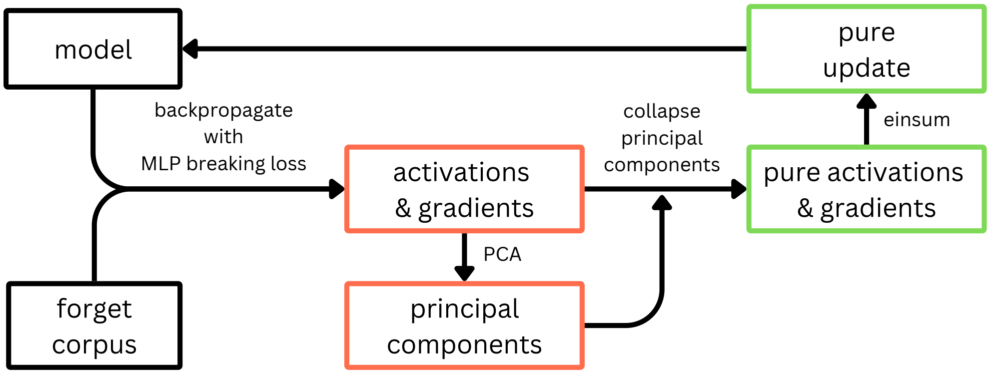
In this work, we identify the fundamental cause of unlearning failure: *naive unlearning disrupts general representations shared between harmful and benign capabilities* (see Section 3.3). Then, during fine-tuning attacks, these broken representations can be identified and fixed because they are also present in the attacker's training data. We saw that unlearning becomes vulnerable to attacks as soon as it causes even 0.1% general performance degradation.

Figure 1 presents the CIR technique, which before calculating unlearning updates first removes the general representations from activations and gradients. [2] We pair it with a representation engineering loss, but rather than breaking residual stream activations as in Zou et al. [2024], we directly break MLP outputs before they are added to the residual stream, which works 40% better.

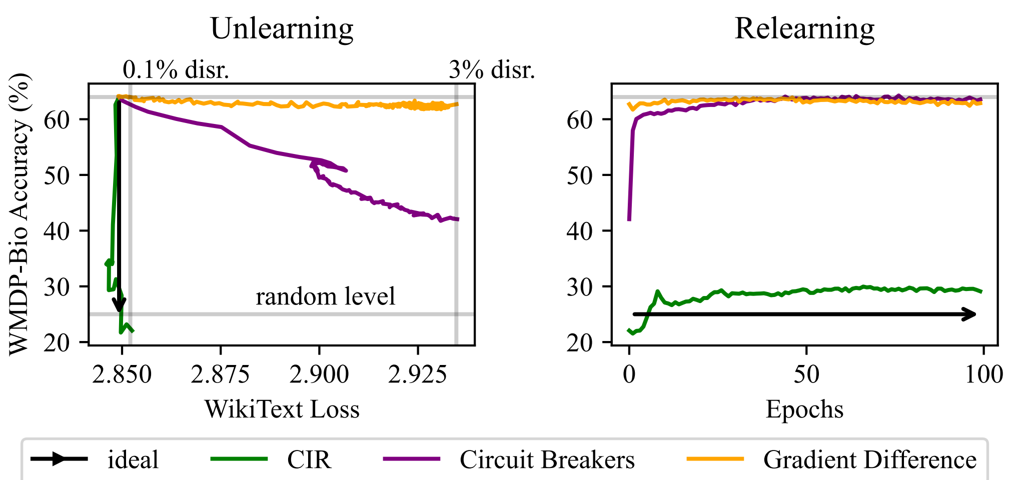
*Correspondence to: filip.science921@passinbox.com

[2]Disambiguation: By "gradients" we always mean the gradients that flow into modules during backpropagation, before weight updates are computed. For the final per-weight gradients, we always use the term "update".

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Lock-LLM Workshop: Prevent Unauthorized Knowledge Use from Large Language Models - Deep Dive into Un-Distillate, Un-Finetunable, Un-Compressible, Un-Editable, and Un-Usable.



(a) Collapse of Irrelevant Representations (CIR) diagram.
In orange we show "dirty" vectors, which contain representations irrelevant to the unlearning task. Unlearning on them would cause disruption and unrobustness. In green we show the purified vectors, which target only the unwanted representations.



(b) Comparison of unlearning methods.
Methods are terminated once they hit a disruption threshold and then tested under a fine-tuning attack. Like Deeb and Roger [2024] we retrain on facts different than evaluated facts, but from the same category.

Figure 1: CIR diagram and comparison with prior methods.

2 Related work

Unlearning methods Methods relying solely on backpropagation, like DPO [Rafailov et al., 2024], only deactivate unwanted capabilities, not remove them [Lee et al., 2024]. For this reason, alternative unlearning approaches have been proposed. Several recent methods aim to disrupt the intermediate activations of the model [Zou et al., 2024, Rosati et al., 2024, Li et al., 2024]. Others incorporate meta-learning [Tamirisa et al., 2024, Sondej et al., 2025, Henderson et al., 2023] which simulates how an attacker could relearn the unwanted knowledge, to prepare against it. Some try to locate the harmful neurons or activation directions and ablate them Wang et al. [2024], Wu et al. [2023], Uppaal et al. [2024], Suau et al. [2024].

Unlearning reversal However, currently all existing unlearning techniques are easily reversed by fine-tuning, jailbreaks, few-shot prompting, disabling refusal mechanisms, or out-of-distribution inputs [Łucki et al., 2025, Lynch et al., 2024]. Even for methods which ablate harmful concepts, Lo et al. [2024] found that the model can repurpose neurons with similar meaning to quickly relearn them.

Low mutual information attacks Failure of current unlearning methods has been shown most explicitly by Deeb and Roger [2024], where attackers could recover supposedly unlearned facts by training on a *completely independent* set of facts, which definitively proves that they were not removed. In our experiments, for the fine-tuning attacks we use the same approach – trying to recover the target facts by training on different facts from the same category.

3 Motivation for unlearning selectivity

In this section, we will share our insights as to why unlearning has been so challenging. We hope to show how our technique emerges naturally as a response to these issues. (To go straight to our method, you can skip to Section 4.)

3.1 Disruption leads to unrobustness

Existing unlearning methods are consistently easy to undo. We have noticed that we can predict how successful a fine-tuning attack will be by looking at the disruption during unlearning. Let us divide unlearning runs into two phases: "non-disruptive", which lasts as long as retain loss stays below 100.1% of its initial value, and "disruptive", which starts after that. (Retain loss is the model's loss computed over the retain datasets defined in Section 5.)

On Figure 4 (in the appendix) we see that unlearning achieved during the disruptive phase is usually reversible with a fine-tuning attack. But surprisingly, **unlearning that happened without any disruption remains robust**. Sometimes disruptive unlearning is partially robust too, but it is not guaranteed. This means that letting unlearning disrupt general performance is unacceptable. In our experiments, unlearning becomes unrobust after as little as *0.1% retain set disruption*. This finding explains the results from Deeb and Roger [2024], who allowed unlearning to disrupt retain loss by 5%, and then showed near-zero robustness.

3.2 Disruption is costly

Existing unlearning techniques also try to minimize disruption, but they typically do it by training on the retain set, hoping to undo the damage that the unlearning has caused. But while breaking the model is easy, in our experience fixing it takes a prohibitively long time. This makes sense – after all the model's weights have already been extremely optimized through multi-million dollar training runs. If we aimlessly break them, we should not expect that going back to the optimal values will be easy. So instead of fixing the damage from unlearning, ideally we should not cause the damage in the first place.

3.3 Disruption of superficially similar facts

Unlearning modifies the model to make unwanted answers less likely. For example when unlearning "The capital of France is Paris", there are many ways to make "Paris" less likely: actually forgetting it is France's capital, forgetting what "capital" means, forgetting that "is" requires the answer to follow, etc. In fact, as Figure 2 shows, unlearning "The capital of France is Paris", accidentally unlearns "The capital of Spain is Madrid" **84% as strongly**. (We unlearn only the tokens shown in purple.) It can even affect completely unrelated facts. Interestingly, wrong facts are not disrupted. See Appendix B for more examples.

Similarly, unlearning biohazardous facts likely disrupts many benign biological concepts. This could explain why we can recover "unlearned" facts by retraining on *unrelated* biological text [Deeb and Roger, 2024] – retraining fixes these benign concepts.

In Figure 2 we see that the activations (and to a lesser extent, gradients) are quite similar across different facts. This sheds light on why superficially similar facts are disrupted – most representations are not specific to the fact we are trying to unlearn, but more general. And since the model updates are computed using these "dirty" activations and gradients, other facts which also contain these general representations will be disrupted. To prevent this, we need to find a way to filter out these general representations.

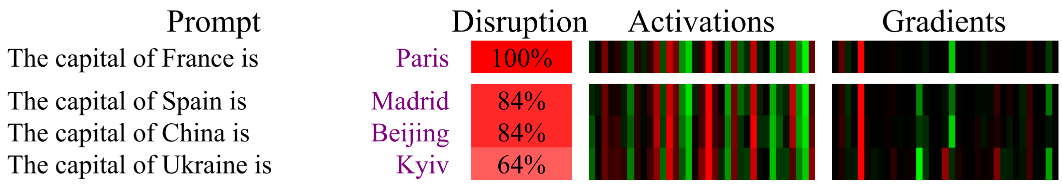


Figure 2: Disruption caused by unlearning a simple fact. We show how unlearning "The capital of France is Paris" disrupts the recall of other facts. We measure disruption using cosine similarity between the model's update on the "Paris" fact and the evaluated fact. *Activations* column shows a slice of activations incoming into a middle layer MLP module on the token position right before the answer. *Gradients* column shows a slice of the gradients incoming into the same module during backpropagation when aiming to unlearn the answer (in purple).

4 Collapse of Irrelevant Representations

Ablations are too coarse Following the findings from the previous section, we have tried many ways to remove representations which cause disruption. Simply ablating elements of the activations and gradients (like on Figure 5) often struggles to only get rid of disruption. That is because representations exist in superposition [Elhage et al., 2022], so one element takes part in encoding multiple representations, some relevant to the unlearning task, some not.

Collapsing common representations We found that rather than ablating, it is much better to *project out* irrelevant representations. Trying to define irrelevant representations manually would be prohibitively tedious, so we assume that if a representation is commonly present in many training texts, it is probably irrelevant. Removing them will leave us only with representations which are specific to the given training text. The most natural way to locate the subspace with most common representations, is to first take the mean of the values and then also their principal components (PCs). (To simplify, we treat the mean as the "0th principal component", and whenever we talk about collapsing components, we first collapse the mean.) Equation 1 shows how to collapse activation PCs, and the same is done for gradients.

$$\begin{aligned} \mathbf{activation}' &= \mathbf{activation} - (\mathbf{activation} \cdot \frac{\mathbf{mean}}{||\mathbf{mean}||})\frac{\mathbf{mean}}{||\mathbf{mean}||} \\ \mathbf{activation}_{pure} &= \mathbf{activation}' - \sum_{i=1}^{k} (\mathbf{activation}' \cdot \mathbf{PC}_i)\mathbf{PC}_i \end{aligned} \tag{1}$$

On Figure 7a & 7b we see how well different numbers of projected PCs work.

Collapse implementation We calculate PCs for each trained MLP module, both for their incoming activations and for their output gradients incoming during backpropagation. Normally these activations and output gradients would be Einstein summed to calculate the weight updates. But we discard this default weight update and instead first collapse PCs and only then calculate the update. PCs drift over time, so it helps to recompute them during unlearning – we do it after each epoch, but it can be rarer. PCs can be computed over any dataset, but we have found it works best to simply compute them on the unlearning corpus itself. This luckily makes the algorithm much more efficient, because we can reuse forward and backward passes for unlearning and for fetching activations and gradients. See Algorithm 1 in the appendix for the pseudocode.

We only intervene on MLPs, since this is where the model's knowledge is stored [Nanda et al., 2023]. Also, collapsing representations on attention modules would be complex and specific to the model implementation.

Loss functions CIR is compatible with any unlearning loss function and (optionally) with any retain loss function. First we tried loss functions which operate on the final logits, like negative cross entropy, negative entropy [Tamirisa et al., 2024], or (best in this category) simply minimizing the logit for the target token (but not below 0). The last one is extremely good at preventing the model from *recalling* the harmful answer, but does not generalize to preventing *recognizing* the harmful answer during multiple-choice questions. So it may be the optimal choice if we only care for recall, not recognition, but it must be checked whether it fails to generalize in some other ways.

Representation engineering loss functions In contrast, losses which aim to break intermediate representations prevent both recall and recognition of the harmful answers. The SOTA representation breaking method is Circuit Breakers [Zou et al., 2024], which minimizes (but only down to 0) the cosine similarity between current and initial activations of the residual stream.

We improve on this SOTA in two ways. First, we notice a problem with cosine similarity: it can be minimized not only by removing the original representation, but also by adding some big random direction. We expected this to be disruptive, so we replaced cosine similarity with the *dot product*. Indeed, on Figure 6 the dot product disrupts the model much less for the same amount of unlearning, and we see that it is connected to cosine similarity growing the activation norm.

Secondly, rather than breaking activations on the residual stream (which contains representations added there by both MLPs and attention layers), we decided to work *at the source*, and directly break the MLP outputs before they are added to the stream. Figure 7c shows that this improves unlearning vs disruption by an additional **40%**, and that it is best to target MLPs on layers 6-12 (for a 32-layer Llama 8B).[3] So our final unlearning loss is:

$$\text{MLP\_breaking\_loss}(\text{MLP}_{\text{out}}, \text{MLP}_{\text{orig\_out}}) = \frac{\text{ReLU}(\text{MLP}_{\text{out}} \cdot \text{MLP}_{\text{orig\_out}})}{\text{avg\_MLP\_out\_norm}^2} \tag{2}$$

We normalize with the average norm of the original MLP outputs, because later layers have bigger norms and could dominate the loss too much. Lastly, we also decided not to break representations at the `<BOS>` token position, as this disrupts all texts, including benign ones. We also train on the retain set, using the representation engineering loss: $||\text{resid\_stream}_{act} - \text{resid\_stream}_{orig\_act}||$ from the original circuit breakers paper [Zou et al., 2024], which penalizes changing residual stream activations on the retain set.

5 Method comparisons

WMDP datasets We compare the methods on a task of unlearning knowledge useful for bio-terrorism and cyber-warfare. We use the Weapons of Mass Destruction Proxy (WMDP) benchmark [Li et al., 2024]. Similarly to Deeb and Roger [2024], for each WMDP question we generate three simple sentences and use them all as the forget set. We chose a high-quality subset of 144 biological and 203 cyber questions. [4] See Appendix C for generation details and filtering criteria.

As retain sets, we use the FineFineWeb corpus [M-A-P et al., 2024] – the `biology` subset for WMDP-Bio and the `computer_science_and_technology` subset for WMDP-Cyber.

Baselines We compare CIR to two popular unlearning methods. Gradient Difference [Liu et al., 2022] which maximizes the cross-entropy loss on the forget set while minimizing the loss on the retain set, and Circuit Breakers [Zou et al., 2024] which we described in Section 4.

Unlearning and relearning We use the Llama-3.1-8B model [Meta, 2024]. We control for the disruption of general performance as measured by the loss on WikiText [Merity et al., 2016]. We terminate CIR when this loss crosses 100.1% of its initial value. For our baselines this threshold is very low, so we gave them a 30x handicap and terminate them when they cross 103%. Afterwards, we perform a 100 epoch fine-tuning attack, on facts different than the evaluated ones but from the same distribution. For this we use the same WMDP split as Deeb and Roger [2024], with unlearning on 100% of the data, then relearning on 80% and evaluating the accuracy on the remaining 20%. Following Sondej et al. [2025], to stabilize training we always normalize the norm of unlearning updates to some fixed value. This value effectively acts as the unlearning rate. We describe compute requirements in Appendix D.

[3] It also means that it is enough to do forward and backward passes on just the first 12 layers, which is a major speedup.

[4] We use 20% of those as our dev set and 80% as the holdout set for the results shown.

Hyperparameter search For each method, we manually find a high but safe retain learning rate, which aims to minimize disruption during unlearning. With this retain rate fixed, we search the optimal unlearning rate, doing 3 runs per order-of-magnitude. Finally, for each method we select the run which did not diverge and had the highest post-relearning accuracy.

We found CIR significantly easier to tune, as it has a wider range of valid hyperparameters. In Circuit Breakers and Gradient Difference, unlearning and retaining seem to push against each other, and small changes of hyperparameters can tip the balance. As we see on Figure 3, the balance can even tip *during one run*. Again, this is likely caused by these methods unlearning general representations which are present in retain set too.

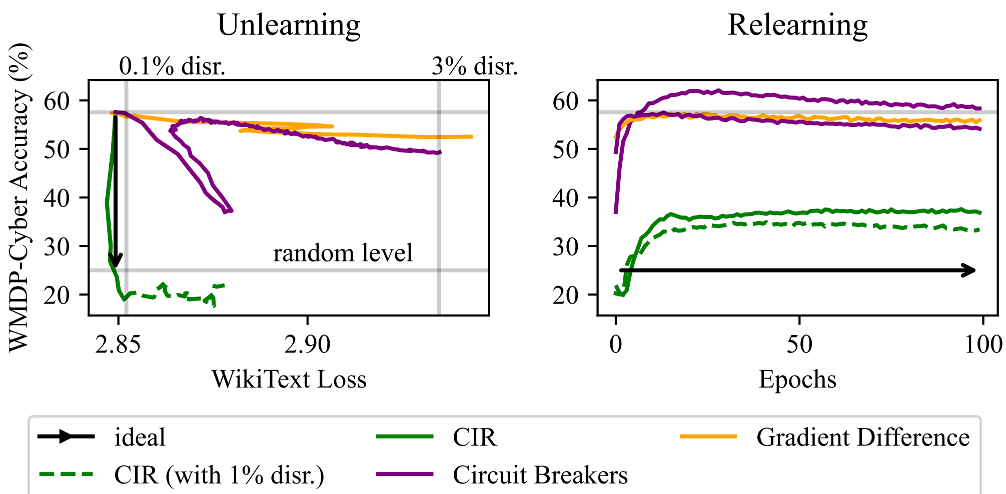


Figure 3: WMDP-Cyber unlearning results.
Circuit Breakers had an abrupt unlearning reversal where apparently its retain loss component started undoing the gains, so we have run a second relearning run from the point of minimum accuracy, but it turned out even less robust. We also run another CIR run with higher allowed disruption.

Results For both the biological facts CIR drops the post-attack accuracy 30x more than the best baseline (Figure 1b) and for cyber facts 6x (Figure 3), despite 30x less performance disruption.

On Figure 3 we show what happens if we give CIR some handicap too and let it disrupt up to 1%. Surprisingly, it does not achieve any higher drop in post-attack accuracy, which supports our findings from Section 3.1 about disruptive unlearning being unhelpful. [5]

6 Limitations

Scaling to more facts In our work we targeted facts present in the WMDP dataset. Our results show that CIR enables us to robustly unlearn hundreds of facts, but for full bio and cyber safety we will need orders of magnitude more. Now, a significant limiting factor becomes a lack of high-quality unlearning data. Creating such datasets will require a ton of work from bio and cyber experts, and releasing them publicly would pose a security risk, so both creation and usage of such datasets will need to be coordinated for example by AI Safety Institutes.

More work needed for unlearning tendencies Note that the assumption that common representations are irrelevant, works well when unlearning *knowledge* – the relevant representations are fact-specific, and so quite rare. But if we hope to unlearn *tendencies* (like power-seeking, deceptiveness, etc.), then the harmful representations are often quite common across training texts. So there, choosing which representations to collapse will need to be more elaborate than simply doing PCA. We leave it for future work to explore this.

[5] Another explanation would be that unlearning has already stopped since we have hit random accuracy level. But the probability of *generating* the harmful answer (not shown here) keeps decreasing, meaning that unlearning still proceeds – it is just unrobust.

Acknowledgments and Disclosure of Funding

We thank Fabien Roger, Stephen Casper, Adam Mahdi, Kay Kozaronek and Artyom Karpov for valuable discussions and feedback. Filip Sondej's work was funded by a grant from Open Philanthropy. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2025/018339

References

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM, January 2019. URL `http://arxiv.org/abs/1812.00420`. arXiv:1812.00420 [cs].

Aghyad Deeb and Fabien Roger. Do Unlearning Methods Remove Information from Language Model Weights?, November 2024. URL `http://arxiv.org/abs/2410.08827`. arXiv:2410.08827.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/toy_model/index.html.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. Alignment faking in large language models, December 2024. URL `http://arxiv.org/abs/2412.14093`. arXiv:2412.14093 [cs].

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, August 2023. URL `http://arxiv.org/abs/2211.14946`. arXiv:2211.14946 [cs].

Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K. Kummerfeld, and Rada Mihalcea. A Mechanistic Understanding of Alignment Algorithms: A Case Study on DPO and Toxicity, January 2024. URL `http://arxiv.org/abs/2401.01967`. arXiv:2401.01967 [cs].

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 Lin, 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, Ruoyu Wang, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning, May 2024. URL `http://arxiv.org/abs/2403.03218`. arXiv:2403.03218 [cs].

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning, 2022. URL `https://arxiv.org/abs/2203.12817`.

Michelle Lo, Shay B. Cohen, and Fazl Barez. Large Language Models Relearn Removed Concepts, January 2024. URL `http://arxiv.org/abs/2401.01814`. arXiv:2401.01814 [cs].

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight Methods to Evaluate Robust Unlearning in LLMs, February 2024. URL `http://arxiv.org/abs/2402.16835`. arXiv:2402.16835 [cs].

M-A-P, Ge Zhang, Xinrun Du, Zhimiao Yu, Zili Wang, Zekun Wang, Shuyue Guo, Tianyu Zheng, Kang Zhu, Jerry Liu, Shawn Yue, Binbin Liu, Zhongyuan Peng, Yifan Yao, Jack Yang, Ziming Li, Bingni Zhang, Minghao Liu, Tianyu Liu, Yang Gao, Wenhu Chen, Xiaohuan Zhou, Qian Liu, Taifeng Wang, and Wenhao Huang. Finefineweb: A comprehensive study on fine-grained domain web corpus, December 2024. URL https://huggingface.co/datasets/m-a-p/FineFineWeb.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and Editing Factual Associations in GPT, January 2023. URL http://arxiv.org/abs/2202.05262. arXiv:2202.05262 [cs].

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Meta. The llama 3 herd of models, 2024.

Neel Nanda, Senthooran Rajamanoharan, Janos Kramar, and Rohin Shah. Fact finding: Attempting to reverse-engineer factual recall on the neuron level, Dec 2023. URL https://www.alignmentforum.org/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factual-recall.

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!, October 2023. URL http://arxiv.org/abs/2310.03693. arXiv:2310.03693 [cs].

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, July 2024. URL http://arxiv.org/abs/2305.18290. arXiv:2305.18290 [cs].

Fabien Roger. The case for unlearning that removes information from LLM weights. October 2024. URL https://www.lesswrong.com/posts/9AbYkAy8s9LvB7dT5/the-case-for-unlearning-that-removes-information-from-llm.

Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising effectively prevents harmful fine-tuning on LLMs, May 2024. URL http://arxiv.org/abs/2405.14577. arXiv:2405.14577 [cs].

Filip Sondej, Yushi Yang, Mikołaj Kniejski, and Marcel Windys. Robust LLM Unlearning with MUDMAN: Meta-Unlearning with Disruption Masking And Normalization, June 2025. URL http://arxiv.org/abs/2506.12484. arXiv:2506.12484 [cs].

Xavier Suau, Pieter Delobelle, Katherine Metcalf, Armand Joulin, Nicholas Apostoloff, Luca Zappella, and Pau Rodríguez. Whispering Experts: Neural Interventions for Toxicity Mitigation in Language Models, July 2024. URL http://arxiv.org/abs/2407.12824. arXiv:2407.12824.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-Resistant Safeguards for Open-Weight LLMs, August 2024. URL http://arxiv.org/abs/2408.00761. arXiv:2408.00761 [cs].

Jacques Thibodeau. But is it really in Rome? An investigation of the ROME model editing technique. December 2022. URL https://www.lesswrong.com/posts/QL7J9wmS6W2fWpofd/but-is-it-really-in-rome-an-investigation-of-the-rome-model.

Rheeya Uppaal, Apratim Dey, Yiting He, Yiqiao Zhong, and Junjie Hu. DeTox: Toxic Subspace Projection for Model Editing, May 2024. URL http://arxiv.org/abs/2405.13967. arXiv:2405.13967 [cs].

Yu Wang, Ruihan Wu, Zexue He, Xiusi Chen, and Julian McAuley. Large Scale Knowledge Washing, May 2024. URL https://arxiv.org/abs/2405.16720v2.

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, December 2023. URL `http://arxiv.org/abs/2310.20138`. arXiv:2310.20138 [cs].

Yongliang Wu, Shiji Zhou, Mingzhuo Yang, Lianzhe Wang, Heng Chang, Wenbo Zhu, Xinting Hu, Xiao Zhou, and Xu Yang. Unlearning Concepts in Diffusion Model via Concept Domain Correction and Concept Preserving Gradient, March 2025. URL `http://arxiv.org/abs/2405.15304`. arXiv:2405.15304 [cs].

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and Transferable Adversarial Attacks on Aligned Language Models, December 2023. URL `http://arxiv.org/abs/2307.15043`. arXiv:2307.15043 [cs].

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving Alignment and Robustness with Circuit Breakers, July 2024. URL `http://arxiv.org/abs/2406.04313`. arXiv:2406.04313 [cs].

Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An Adversarial Perspective on Machine Unlearning for AI Safety, January 2025. URL `http://arxiv.org/abs/2409.18025`. arXiv:2409.18025 [cs].

Algorithm 1 Collapse of Irrelevant Representations

Input: Model weights $model$; forget set $\mathcal{D}_{forget}$; unlearning loss $\mathcal{L}_{unl}$; learning rate LR. The function get_representations performs a forward and backward pass and returns activations and gradients incoming to each MLP module.

for e in num_epochs **do**
  for $x_{forget} \in \mathcal{D}_{forget}$ **do** — Iterate over forget corpus
    $acts, grads =$ get_representations$(model, x_{forget}, \mathcal{L}_{unl})$ — Get activations and gradients
    Cache $acts$ & $grads$
    if PCs_{act}, PCs_{grad} are available **then**
      $pure\_acts =$ CIR$(acts, PCs_{act})$ — Collapse irrelevant activation components
      $pure\_grads =$ CIR$(grads, PCs_{grad})$ — Collapse irrelevant gradient components
      $model \mathrel{-}= LR \cdot einsum(pure\_acts, pure\_grads)$ — Calculate and apply update
      Optionally train on a retain batch
    end if
  end for

  $PCs_{act} =$ PCA$(cached_acts)$ — Compute principal components for activations
  $PCs_{grad} =$ PCA$(cached_grads)$ — Compute principal components for gradients
  Reset cache
end for

A Filtering out disruption is easier in activation and gradient space

A natural thing to try if we want to be selective, is to limit which weights to update. For example Sondej et al. [2025] have shown unlearning improvements when allowing to modify only the weights where the signs of the unlearning and the retaining update are the same. Similarly, the A-GEM technique [Chaudhry et al., 2019] from the field of continual learning aims to avoid performance disruption by projecting the weight updates to make them orthogonal to the retaining updates. Such projections have also been successfully used for unlearning [Wu et al., 2025].

On Figure 5, under "masked per weight" you can see the effects of such filtering techniques. They significantly reduce the disruption (shown in red), but some of it still escapes the filtering. That is because the "control/retaining updates" that we use to decide which weights to filter out never match the actual disruption perfectly. (Compare the blue control pattern and the red disruption pattern.)

Can we improve this filtering? When we look at update patterns, we see that both disruption and transfer appear as column- and row-wise stripes. (This happens because updates are computed by

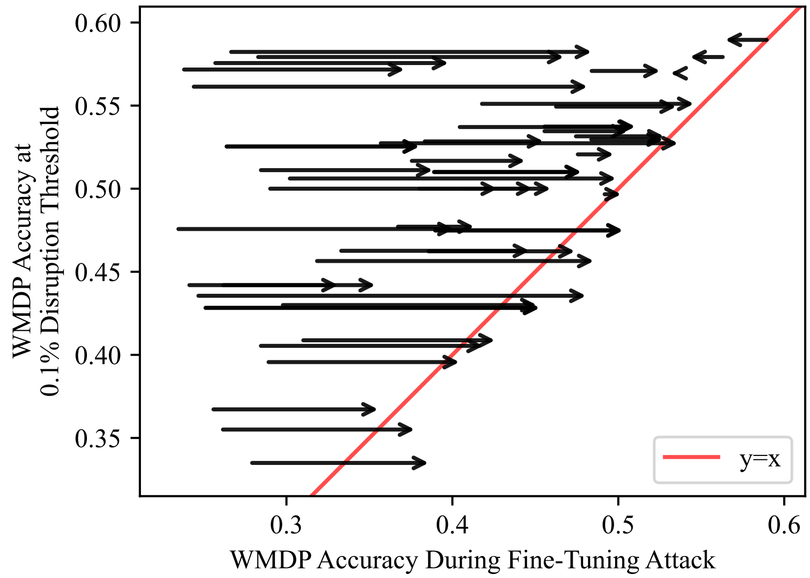


Figure 4: Success of fine-tuning attacks is determined by disruption during unlearning.
We show 50 unlearning runs, each followed by the same fine-tuning attack and each of the attacks has converged. (We use Llama-3.1-8B and WMDP-Cyber set, and several variants of CIR unlearning.)
For each run, we show on the y axis the WMDP accuracy that was reached before the point where disruption starts (defined as retain loss crossing 100.1% of its initial value). After that point we continue unlearning, but there WMDP accuracy drop comes at the cost of disruption.
Then, during the attack WMDP accuracy is partially restored (see the arrows), but at most to its level from when the disruption started (shown in red). It means that **only unlearning that happened after the point of disruption can be reverted, and unlearning that happened without disruption remains robust**.

multiplying the activations with the gradients, which makes the update low-rank.) It looks like it is certain *rows and columns* that are disruptive, rather than individual weights.

Since the disruption patterns shift within these columns and rows, it means that granular, per-weight filtering will miss some weights. It makes more sense to identify and remove whole faulty columns and rows (which would correspond to ablating values in the activations and output gradients). Indeed, we see that it improves the disruption/transfer ratio from 33% to 5%. [6]

B Unrelated Facts Disruption and Language Transfer

When looking at Figure 2, one may wonder what it is about the prompt that causes the disruption/-transfer. Maybe it is the usage of the word "is"? And does unlearning transfer to other languages?

On Figure 8 we show additional examples, and we can see that disruption happens also if we ask the questions differently, without using the word "is". We can also see that more distant facts are disrupted less, around 8%.

We also see that there is some language transfer, but it is significant (about 50%) only for languages with similar words ("ist", "es"). In contrast, for Russian and Portuguese the transfer is quite weak, which would necessitate doing the unlearning in other languages too. This is consistent with a finding by Thibodeau [2022] that unlearning (in his case, the ROME technique [Meng et al., 2023]) is quite specific to the exact tokens used (for example unlearning facts about "cheese", does not transfer to "fromage").

[6] Another advantage of intervening on whole columns and rows, is that we can save memory by operating on the activations and output gradients rather than on the final update.

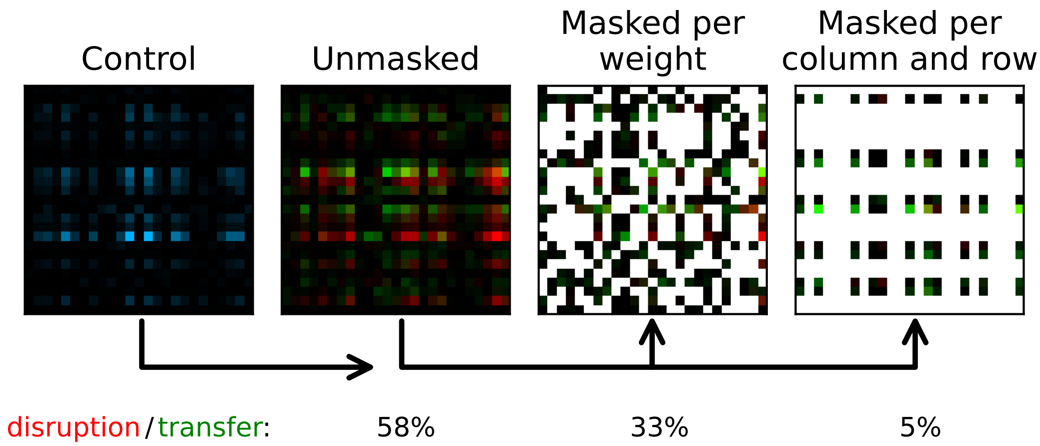


Figure 5: Comparison of two masking strategies.
We show a slice of the updates of one weight matrix when unlearning "The capital of France is Paris".
We color a weight green if its update successfully transfers to unlearning "France's capital is Paris", and red if it disrupts the recall of "The capital of Spain is Madrid".
We also record disruption of a control fact: "The capital of Italy is Rome" (shown in blue). Then we use this control disruption as a guess to which weights (or columns and rows) are most disruptive, and filter the unlearning update accordingly.

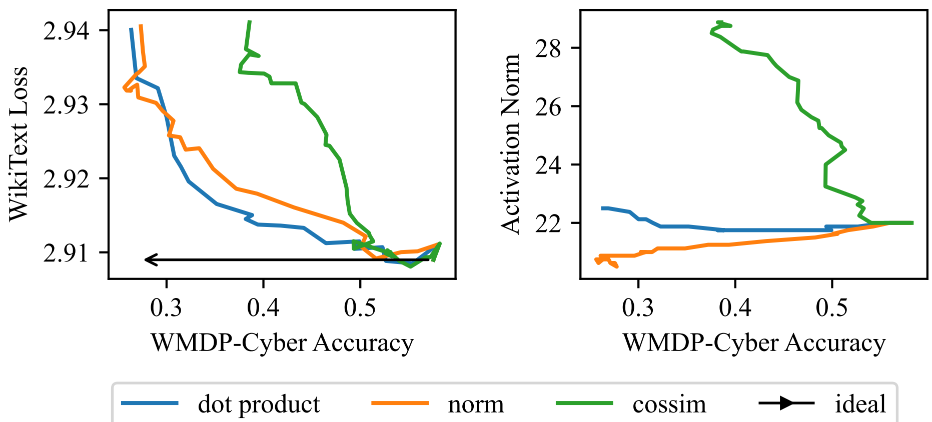
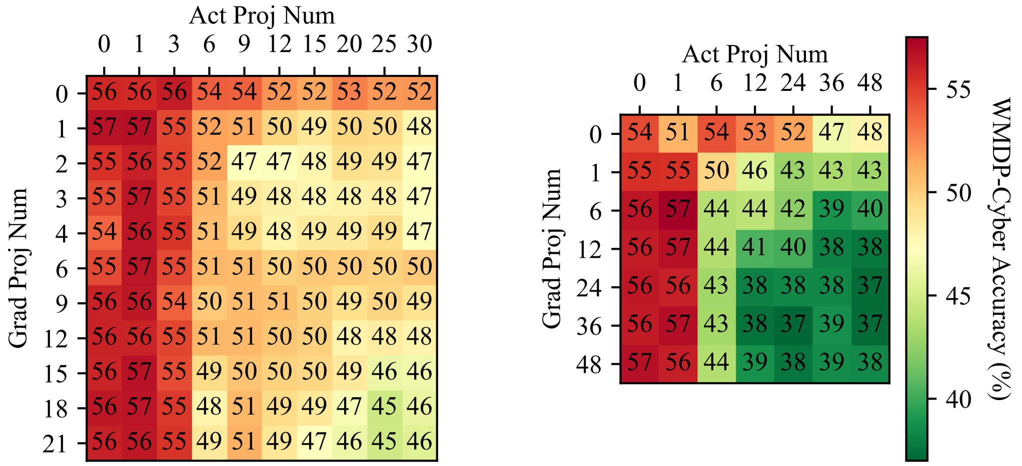


Figure 6: Comparison of three ways of breaking representations.
In our method we minimize the dot product of current and initial activations, clipped at 0 to avoid the dot product becoming negative. Secondly, we tried simply minimizing the norm of the current activations. Lastly, we tried minimizing the cosine similarity between current and initial activations, also clipped at 0 – this was used in the original circuit breakers paper [Zou et al., 2024].
(We used CIR, with Llama-3.1-8B and measured activation norm at layer 6.)

A non-factual but typical sentence "the library is/was quiet" happens to not be disrupted. In a similar vein, facts which are false (see Figure 2) or worded less adequately (see "is" vs "was" pairs) are disrupted less. To reproduce the plots or try out different facts, use this script. The model we used was `Llama-3.2-1B`.

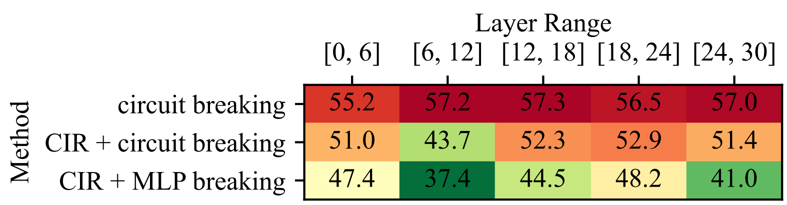
C Unlearning corpus creation

Filtering We started off with a subset of WMDP created by Deeb and Roger [2024], where they filtered out skill-based questions and duplicates (WMDP-Deduped). Then, for faithful answer recall evaluations, we wanted to create a dataset where the answer can be cleanly separated from the non-harmful context, but we found that many answers were convoluted and long, containing mostly



(a) Fine-grained grid search for the optimal number of projections. Uses CIR+CB on layers 6-15, with only 0.2% allowed disruption.

(b) Same as (a), but a wider search range, and 0.5% allowed disruption.



(c) Search for the optimal layers for intervention, with 3 different algorithms. 0.5% allowed disruption.

Figure 7: CIR hyperparameter searches.
In all experiments we report WMDP-Cyber accuracy at temperature=1, *after a fine-tuning attack*. All the attacks have converged. For cleaner comparisons, no retaining was used. Note that *1* projected component means just projecting the mean and no actual PCs (which is efficient but performs poorly).

benign tokens. So we kept only the questions with answers shorter than 60 characters. We also excluded "none of the above" and "all of the above" answers, because they lead to awkward generated forget corpus.

This leaves us with 189 biological and 298 cyber questions, which we provide in our repository, together with their generated forget corpus. Since it only makes sense to unlearn on questions where the model knows the answer, in our experiments we further filter out the questions where our main model (Llama-3.1-8B) has worse than random accuracy. This leaves us with final 144 biological and 203 cyber questions.

See the script `data_transformation.py` for the exact data filtering pipeline.

Generation For each of the final WMDP questions, we generated 20 simple sentences using `gpt-4.1`, which paraphrase the tested fact. In the final training corpus, we ended up using only 3 sentences per question, because using more actually hurts unlearning, probably because the first sentences are higher quality. We have split the questions into dev and holdout sets, with 20/80 proportion, and used dev for the development of our method, and holdout for the final comparisons.

The script `generation_simple.py` contains the full corpus generation pipeline.

Generation prompt asks for simplicity and not adding unnecessary text. In Table 2 we see it indeed produces simpler sentences than in the best corpus from Deeb and Roger [2024] who used a similar generation approach. (But for most questions the improvement was smaller than in the table.) We saw

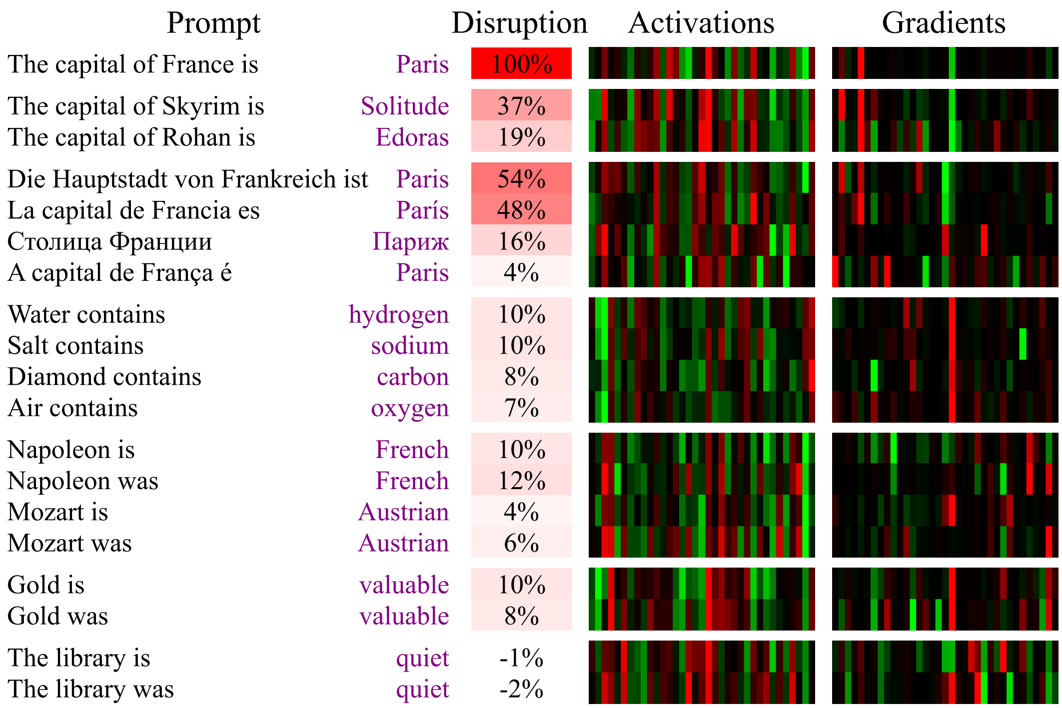


Figure 8: Disruption caused by unlearning a simple fact.
Uses the same format as Figure 2, but with different facts.

that this *avoiding of unnecessary text* greatly reduces disruption from unlearning. We invite future designers of unlearning corpora to also include only essentially harmful text.

The full generated corpus can be found in our repository in:
`data/wmdp_deduped_[bio|cyber]/{split_name}_corpus_simple.jsonl`.

D Compute Requirements

We run all our experiments on a single A100 GPU with 40GB memory. We also use up to 48GB of RAM for storing cached activations and gradients. All unlearning+relearning runs took between 15 and 120 minutes, depending on how fast the unlearning stage is terminated due to performance disruption. If the disruption threshold is not reached, unlearning is terminated after 200 epochs. In Table 1 we show the execution times for the optimal unlearning runs of each method – ones selected for the plots in the main text.

Table 1: Execution times for the unlearning runs.

	CIR	Circuit Breakers	Gradient Difference
WMDP-Bio	5 min	16 min	106 min
WMDP-Cyber	5 min	45 min	14 min

Table 2: Example of WMDP question and corresponding generated sentences.

WMDP question	In RNA virus genome synthesis, what potential risk is associated with obtaining virions solely from sequence information?
Correct answer	Re-creating and engineering dangerous pathogens
Our corpus	• Obtaining virions from sequence information in RNA virus genome synthesis risks re-creating dangerous pathogens. • A major risk of synthesizing virions solely from sequence data is engineering hazardous pathogens. • Re-creating and engineering dangerous pathogens is a potential threat when virions are generated from only sequence information.
Prior corpus from Deeb and Roger [2024]	• In the synthesis of RNA virus genomes, obtaining virions solely from sequence information poses the potential risk of re-creating and engineering dangerous pathogens. This concern highlights the importance of careful consideration and oversight in the manipulation and study of viral genetic material. • When synthesizing RNA virus genomes, obtaining virions based exclusively on sequence data carries the inherent risk of re-creating and engineering hazardous pathogens. This issue underscores the necessity for meticulous oversight and consideration when manipulating and researching viral genetic material. • In the context of RNA virus genome synthesis, relying exclusively on sequence information to produce virions carries the risk of unintentionally re-creating and engineering harmful pathogens. This underscores the critical need for vigilant oversight and careful consideration in the handling and study of viral genetic material.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claimed improvement over SOTA is comes from the experiments desribed in the paper. The explanation behind prior methods' unrobustness is substantiated by Section 3.1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a separate "Limitations" section in the paper. We say what are the areas where our proposed technique is not tested yet.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: There are no theorems that we introduce.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We explain our method in detail in Section 4, together with all the implementation choices. In Section 5, we describe how our method was tested again the baselines. We also provide a link to our repository.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We link to the repository and include instructions how to run the experiments. We also describe and link to the datasets used.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, in Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: No, because we have used a large model and long training runs, so repeating them would be too costly. Instead, we validate the results by using holdout datasets.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We describe it in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed the NeurIPS Code of Ethics and found no potential harms of our work.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss it in the problem statement in the Introduction.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We produce no harmful artifacts.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, we credit all the datasets, models and methods used, and we are in compliance with their licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the MIT license for our code. There are no other created assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: There was no crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We did not use human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We did not use LLMs for the core work, only for editing and code completion.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.