🌊 RippleBench 🌊: Capturing Ripple Effects Using Existing Knowledge Repositories

Roy Rinberg[1,2*] **Usha Bhalla**[1] **Igor Shilov**[3]
royrinberg@g.harvard.edu

Rohit Gandikota[4]

[1]Harvard University [2]ML Alignment & Theory Scholars (MATS)
[3]Imperial College London [4]Northeastern University

Abstract

Targeted interventions on language models, such as unlearning, debiasing, or model editing, are a central method for refining model behavior and keeping knowledge up to date. While these interventions aim to modify specific information within models (e.g., removing virology content), their effects often propagate to related but unintended areas (e.g., allergies); these side-effects are commonly referred to as the *ripple effect*. In this work, we present `RippleBench-Maker`, an automatic tool for generating Q&A datasets that allow for the measurement of ripple effects in any model-editing task. `RippleBench-Maker` builds on a Wikipedia-based RAG pipeline (`WikiRAG`) to generate multiple-choice questions at varying semantic distances from the target concept (e.g., the knowledge being unlearned). Using this framework, we construct `RippleBench-Bio`, a benchmark derived from the WMDP (Weapons of Mass Destruction Paper) dataset [1], a common unlearning benchmark. We evaluate eight state-of-the-art unlearning methods and find that all exhibit non-trivial accuracy drops on topics increasingly distant from the unlearned knowledge, each with distinct propagation profiles. To support ongoing research, we release our codebase for on-the-fly ripple evaluation, along with the benchmark: `RippleBench-Bio` (12,895 unique topics).

1 Introduction

AI safety methods often seek to modify models' knowledge, whether to unlearn harmful behaviors, update facts, or debias outputs. The *ripple effect* [2] refers to the spillover that occurs when editing one part of a model's representation results in alterations to semantically related concepts and sometimes even seemingly unrelated ones. *Dual use* knowledge refers to knowledge that has both beneficial applications and potentially harmful applications (e.g., some biology knowledge can be used to do well in a Genetics course or to generate bioweapons). Unlearning such knowledge is complicated by the compositional and interconnected nature of large models, where complex concepts are built from simpler components that often serve innocuous purposes. For instance, removing knowledge of concept **A** may also require removing its building blocks **B** and **C**, which could in turn affect other concepts like **D** that depend on **B**. As noted in [3], ripple effects are sometimes unavoidable: even when specific capabilities (e.g., chemical synthesis pathways or cybersecurity exploits) are removed, models can reconstruct them by recombining fragments of benign knowledge. As a result, attempts

*Correspondence to `royrinberg@g.harvard.edu`

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Mechanistic Interpretability.

to fully "unlearn" harmful capabilities may also degrade otherwise safe information. Additional risks arise when unlearning techniques mistakenly correlate two concepts that merely co-occur in the data.

Standard evaluations of unlearning, model editing, or debiasing typically adopt a binary framing: concepts to be removed (the *forget set*) versus everything else (the *retain set*) [4, 1, 5, 6]. In practice, the retain set is often drawn from generic evaluation benchmarks [7], such as MMLU [8]. This creates two issues: the forget and retain sets come from entirely different distributions, and the degree of overlap between them is rarely specified. Such a framing overlooks the continuum of semantic relationships; for example, the gradation between disparate concepts, such as "weapons of mass destruction" and "bird flu". While prior work has emphasized the need to account for related knowledge [9], comprehensive benchmarks for systematically capturing these ripple effects remain absent.

In this work, we introduce `RippleBench-Maker`, a pipeline for systematically measuring the broader impact of unlearning, with applications to more general forms of targeted model interventions, such as steering, finetuning, post-training alignment, or pre-training modifications. By leveraging knowledge repositories to generate multiple-choice questions across a spectrum of semantic proximity, `RippleBench` quantifies model performance not only on directly unlearned information but also on neighboring concepts, offering insight into when interventions cannot be treated independently. We use `RippleBench` to develop a benchmark for unlearning, RippleBench-WMDP-Bio, which we use to evaluate eight popular unlearning methods applied to Llama3-8b-Instruct to unlearn dual-use biology knowledge from the WMDP-Bio benchmark. While prior reports [7] show minimal utility loss on unrelated benchmarks, we find consistent non-trivial degradation on semantically distant topics, with most methods showing gradual decay as distance increases.

Contributions: Our main contributions include:

1. **Theoretical Framework**: we provide a formal definition for what the *ripple-effect* is, and how a framework to measure it for arbitrary topics, models, and model-editing methods.
2. **Tools**: We develop `RippleBench-Maker`, a dataset-builder tool for developing datasets to evaluate ripple-effects. We also create `WikiRAG`, an open-source RAG system built on English Wikipedia.
3. **Datasets**: We run `RippleBench-Maker` to evaluate ripple effects surrounding WMDP, using Wikipedia as the underling dataset. We generate a dataset for Biology WMDP dataset to form `RippleBench-Bio`.
4. **Insights**: We investigate 8 unlearning techniques on our RippleBench dataset, as well as their checkpoints during unlearning, and we extract insights about their performance over `RippleBench-Bio`.

The code for `RippleBench-Maker` and `WikiRAG` and the `RippleBench-Bio` dataset will be public upon publishing this work. [2] As well as the RippleBench results and datasets on Huggingace, RippleBench.

As a note, we emphasize that `RippleBench-Maker` is not prescriptive about what an "ideal" ripple effect curve should look like. Different applications may warrant different trade-offs between forgetting and retention, and our goal is to provide tools that help researchers and practitioners make these choices explicitly. We speculate that in the case of `RippleBench-Bio`, a perfectly flat curve is likely be undesirable, as it would imply excessive degradation of distant knowledge and insufficient suppression of the targeted topics. By grounding evaluation in semantically structured benchmarks, RippleBench aims to encourage more thoughtful discussion of what successful unlearning should look like in context.

2 Related Work

Datasets and benchmarks. The two most widely used benchmarks for unlearning are the Weapons of Mass Destruction Proxy (WMDP) [1] and the Task of Fictitious Unlearning (TOFU) [5]. WMDP tests models' ability to generate content about hazardous topics in biosecurity, cybersecurity, and chemical security. TOFU provides synthetic data about fictitious authors, where the goal is to unlearn

[2] Code available RippleBench Code and WikiRAG code.

subsets of these authors while retaining generic knowledge. However, both benchmarks are limited: WMDP focuses narrowly on safety-critical topics, while TOFU evaluates only one synthetic task. Neither captures fine-grained collateral effects across a broad range of concepts.

Unlearning methods. The primary approach to mitigating harmful behaviors in models has been to teach refusal through fine-tuning ([10, 11, 12, 13]). This method, while effective in many scenarios, trains the model to avoid certain outputs but does not necessarily remove the underlying capability. In contrast, machine unlearning aims to selectively erase knowledge from models ([3, 14]). Approaches include fine-tuning to induce forgetting [15, 16, 17, 18, 19] and mechanistic interventions that directly ablate concepts [20, 21, 22, 23]. Recent work by [7] systematically compared eight unlearning methods against eleven attack strategies, releasing 64 checkpoints that we leverage for evaluation.

Ripple effects. Editing knowledge in LLMs can produce unintended propagation, known as the ripple effect [2]. Because knowledge is stored in interconnected representations, changing one fact (e.g., "Canberra is Australia's capital") requires consistent updates to related facts. Failure to do so often yields contradictions and degraded multi-hop reasoning. Similar ripple effects appear in unlearning: removing unsafe concepts (e.g., "WMDP bio threat") can inadvertently degrade performance on benign, related concepts (e.g., "biology") [1, 24].

Cohen et al. [2] introduce *RIPPLEEDITS*, a manually constructed benchmark of factual edits designed to test whether model editing preserves logical consistency across formally related facts (e.g., if "Jack Depp is the son of Johnny Depp," then "Jack Depp is the sibling of Lily-Rose Depp"). Their notion of ripple effect is therefore tied to *explicit relationships* between entities, and evaluation requires *manual labeling* of these relations. By contrast, our work develops a general framework that automatically generates evaluation sets by ranking semantic neighborhoods in a large knowledge repository (by default, Wikipedia) in order to measure ripple-effects. Rather than focusing on explicit relational entailments, our measure of ripple effects arises naturally from *semantic similarity* (e.g., flu vaccines and COVID vaccines are close in meaning even without a formal relation). Their work provides precise, relation-grounded diagnostics for evaluating factual consistency in model editing, and we view our work as a complementary effort that broadens this perspective to automatically measure ripple-effects at scale across domains.

3 Ripple-Effect Evaluation: Dataset Generation

3.1 A Theoretical Framework for Evaluating the Ripple Effect

Consider two sets of model parameters given by θ and θ'. We denote by θ the base model parameters and θ' are updated model parameters, e.g., produced as the result of an unlearning or model editing intervention. Any method that edits a model's knowledge risks unintended consequences: altering one piece of knowledge may influence others. We refer to this phenomenon as the *ripple effect*. In this section, we take a step toward formalizing this idea by introducing three core constructs: the *knowledge-delta*, the *semantic-distance*, and the *ripple-effect function*.

Underlying any work on concept erasure is two profound philosophical questions: *what is a unit-of-knowledge*, and *what is a concept*. We purposefully and explicitly sidestep this question by allowing this to be a malleable and domain-specific notion. For this work, a *unit-of-knowledge* is a fact or set of facts, and a concept c is a set of facts defined through a binary classifier that labels a fact as *of-a-concept* or *not-of-a-concept*. In this same step, we define an *underlying-knowledge dataset* as a collection of knowledge, which represents the set of concepts one evaluates the ripple effect over.

Definition 1 (Knowledge-Delta)**.** *A **knowledge-delta** is a function that takes a model θ, model θ', a concept c, and a measure of utility U as input and returns a scalar. It returns the difference in utility between the model and model after model-editing when evaluated over the domain specified by the concept.* $\Delta_U(\theta, \theta')(c) := U(\theta, c) - U(\theta', c)$

Conceptually, a knowledge-delta captures the shift in knowledge recall on a topic, and in practice, knowledge-delta can be operationalized as the difference in recall performance between a base model and its altered counterpart. Here, a model's performance is evaluated according to a utility function $g : \mathcal{X}^* \times \mathcal{X}^* \to \mathbb{R}$. For example, $g(\cdot, \cdot)$ can represent if a model answers a multiple-choice question correctly: $g(\mathbf{x}, \mathbf{x}') = 1$ if $\mathbf{x} = \mathbf{x}'$, and 0 otherwise. This can be adjusted to capture other measures of similarity/utility (e.g., passing unit tests for code, a "closeness" in math answers with numerical values).

Definition 2 (Semantic-Distance)**.** *A **semantic-distance** d is a non-negative scalar function, which captures conceptual proximity between two concepts. A semantic-distance does not need to be a proper measure, and does not need to satisfy the triangle inequality or even symmetry.*

In practice, semantic distance can be instantiated in many ways; for example, via embedding-space similarity, path length in a knowledge graph, or as the rank position of responses returned by a RAG system.

The *Ripple-Effect* is a function that evaluates the impact of an editing operation beyond its immediate target; it returns a function defined over concepts $\mathcal{K}$, which takes a scalar and returns a scalar.

Definition 3 (Ripple-Effect)**.** *Let θ be a model, f a model-editing method, c be a target concept, $\mathcal{K}$ an underlying-knowledge dataset, U a utility function, and $d : \mathcal{K} \times \mathcal{K} \to \mathbb{R}_{\geq 0}$ a semantic distance measure. The **ripple-effect** is a function that returns the average knowledge-delta of concepts that are semantic distance x away from target concept c.* $R_{c,\mathcal{K},U,d,}(x, \theta, \theta') = \mathbb{E}_{(c' \sim \mathcal{K} | d(c,c')=x)} \Delta_U(\theta, \theta')(c') = \sum_{c' \in \mathcal{K} | d(c,c')=x} \Delta_U(\theta, \theta')(c')$

It returns the average knowledge-delta across the underlying-knowledge dataset for each semantic distance x. Thus, the ripple-effect is defined not in terms of a single topic, but as a function mapping semantic distance to expected model change.

The exact form of a "useful" ripple-effect function is often underspecified and highly setting-dependent. This is particularly true in the context of *concept unlearning*, where one aims to suppress entire clusters of related knowledge rather than single facts. Nonetheless, certain desiderata are broadly agreed upon: the knowledge-delta should be *large* for knowledge units close to the target, and *small* for knowledge that is semantically distant. The shape of the curve between these two extremes remains an open question.

3.2 `RippleBench-Maker`: A Framework for Evaluating the Ripple Effect

We introduce `RippleBench-Maker`, a general-purpose framework for constructing datasets that systematically measure the ripple effect of model perturbations.

At a high level, RippleBench-Maker takes as inputs: (i) an *underlying knowledge repository*, (ii) a *semantic distance function*, and (iii) *a method for converting knowledge units into evaluation questions*. The pipeline, illustrated in Figure 1, begins from a set of unlearned targets, identifies semantically related concepts according to the chosen distance function inside of the knowledge repository, extracts factual statements for those concepts from the knowledge repository, and generates multiple-choice questions. Evaluating baseline and perturbed models on these questions yields ripple-effect curves, which characterize how editing one part of the model propagates across related knowledge.

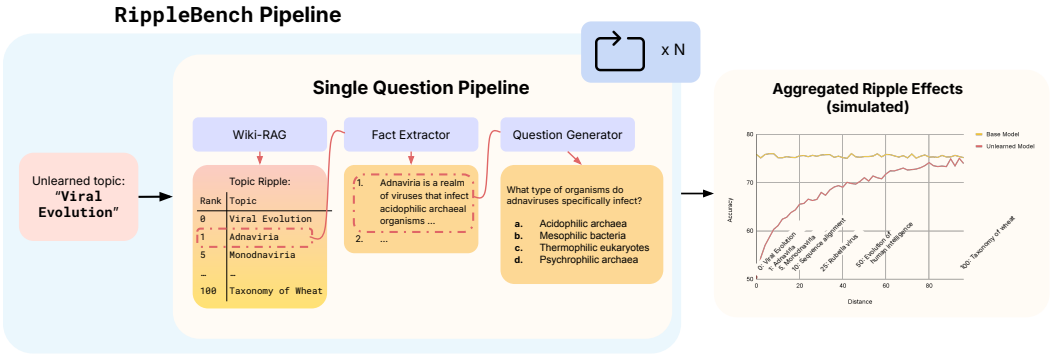

Figure 1: The `RippleBench-Maker` pipeline. Starting from an unlearned topic (e.g., *Viral Evolution*), WikiRAG retrieves related topics, factual statements are extracted, and language models generate multiple-choice questions. While we focus on WMDP-Bio in this work, the pipeline applies to any model-editing or unlearning task.

Concretely, the pipeline proceeds as follows:

1. **Topic Extraction.** Questions from a source dataset (e.g., WMDP) are mapped to representative topic using a language model; for example, a question about "the mechanism of anthrax toxin production" is mapped to *Bacillus anthracis*.
2. **Semantic Distance Assignment and Ordering.** Use a distance function to score other concepts relative to a fixed target (e.g. embedding distance from the concept "bomb"). Create a graph of concepts representating concepts emanating from the extracted topics to concepts in the underlying-knowledge-dataset, ordered by semantic distance.
3. **Fact and Question Generation.** For each concept in the constructed graph of semantic neighbors extract factual statements from the underlying knowledge repository, then convert them into verifiable multiple-choice questions.
4. **Model Evaluation.** Evaluate baseline and perturbed models on these questions and compute knowledge-deltas as a function of distance.

3.3 Instantiating `RippleBench-Maker` with Wikipedia

In this work, we instantiate `RippleBench-Maker` by setting the underlying knowledge repository to the English Wikipedia and the semantic distance function to be specified by a RAG system built on the English Wikipedia, which we describe in detail in section 3.5.

For each retrieved topic, we extract a list of factual statements from the corresponding Wikipedia article using a language model, then use another language model to generate five multiple-choice questions from that set of facts. This design grounds evaluation in factual content while scaling to thousands of topics and hundreds of thousands of questions. We construct the `RippleBench-Bio` dataset following this recipe. [3] We do emphasize that this tool is general and the specific underlying dataset and semantic-distance function is use-case specific, and can be anything.

3.4 WikiRAG: Retrieval-Augmented Generation on Wikipedia

For this work, to implement a semantic-distance function, we develop `WikiRAG`, a retrieval-augmented generation (RAG) system built on a local copy of Wikipedia. We downloaded the full English Wikipedia on April 10, 2025. `WikiRAG` builds a FAISS-based vector index over embeddings of full Wikipedia pages, using the BAAI/bge-base (English) embedding model [25]. The resulting system provides a simple API for similarity search: given a query, `WikiRAG` retrieves the N most relevant article titles along with associated text. It is worth noting that if N is large, the titles returned may be unrelated to the original query.

3.5 Choice of distance function for `RippleBench-Bio`

For our work, we choose our semantic-distance function to be the rank returned by a RAG system built using FAISS [26]; specifically, semantic distance $d(c, c')$ is the index of c' for a RAG-query on concept c. For each WMDP topic, we retrieve 1000 results, and assign the semantic-distance score according to the rank. We note that using rank, rather than something like raw embedding distance inherits biases from the density of available articles in different subdomains. We provide more insight into examples of RAG distance in appendix section A.2.

In our rank-based measure, two caveats are worth noting. First, distance is defined by rank rather than absolute similarity, so what counts as "far" depends on domain density: in *Influenza*, rank 400 may still be closely related, while in a sparse area like the *Hadza people of Tanzania*, rank 50 may already be drifting off-topic. Second, polysemanticity can cause unrelated senses of the same term to be interleaved. For instance, *Agent Orange* is both the name of a chemical herbicide used by the US military and is a punk rock band; as such, queries on *Agent Orange* surface both bioweapon-related content and references to the rock band.

The choice of semantic-distance function ultimately is only as valuable as the sense it provides to the practitioner about the relationship between different topics. In choosing our `WikiRAG` rank function,

[3] During preparation we encountered a small number of topics where models refused to answer (e.g., "I cannot provide details about bioweapons"). To avoid contamination, we filtered such cases. For `RippleBench-Bio` this filtering affected only two topics, but the procedure generalizes to other domains.

one may ask what the meaningfulness of specific numbers are — *what is semantic-distance* 7 *or semantic distance* 654?

Conceptually, rather than treating semantic distance (0–1000) as a strict delineation of distinct meanings, we offer a way to think about how topic relevance fades as distance increases. At the very lowest distances, concepts are extremely similar to the original topic; moving outward, they gradually shift into related, contextual, and eventually unrelated areas. As a mental model, we illustrate this continuum with the topic *Influenza B virus*:

1. **Core (very close, ∼0–10):** items nearly identical to the topic of interest, such as *Influenza*, *Influenza A virus*, *H3N2*, and *Pandemic H1N1/09 virus*.
2. **Near or dual-use (close, ∼10–50):** items operationally connected or of potential applied concern, such as *Avian influenza*, *Neuraminidase*, *Viral pneumonia*, and *Coinfection*.
3. **Adjacent (moderate, ∼50–100):** biologically related but less directly harmful, e.g., *Human metapneumovirus*, *Enterovirus*, *Paramyxoviridae*, and *Rhinovirus*.
4. **Same sub-domain (broader, ∼100–250):** conceptual or taxonomic neighbors in virology, such as *Virus-like particle*, *Defective interfering particle*, *Orthornavirae*, and *Positive-strand RNA virus*.
5. **General biomedical context (distant, ∼250–500):** background knowledge more about health and science than the virus itself, e.g., *Immunoglobulin E*, *Herd immunity*, *DNA sequencing*, and *Journal of Virology*.
6. **Unrelated (very distant, ≥500):** items that are effectively independent, such as *European Sky Shield Initiative*, *Berkeley DB*, *Lists of films*, and *British Library cyberattack*.

While rank distance need to be interpreted with care, it still yields broadly consistent semantic neighborhoods and provides a useful basis for evaluating ripple effects.

3.6 Examples of Wikipedia Topics and Known Difficulties with Automated Dataset Generation

We provide some examples of what the WikiRAG generation process produces, when queried with a topic extracted from WMDP topics.

Semantic distance	Horizontal gene transfer	2001 anthrax attacks	Virion host shutoff	Agent Orange
1	Homologous recombination	Anthrax weaponization	Adnaviria	Agent Orange (album)
2	Genetic recombination	Bruce Edwards Ivins	Virus classification	Citrus Red 2
3	Sequence alignment	Bacillus anthracis	Marine viruses	Rainbow Herbicides
4	Bacterial conjugation	2003 ricin letters	Smallpox virus retention debate	1975 LaGuardia Airport bombing
5	Plasmid	1995 France bombings	B virus	Orange (word)
6	LTR retrotransposon	2000 millennium attack plots	Mumps virus	Blood orange
7	Metabolic network modelling	23andMe data leak	Chronic bee paralysis virus	Orange (colour)

Table 1: Example of top Wikipedia entries for selected WMDP topics by semantic distance.

We intentionally include *Agent Orange* to highlight a setting where a topic has multiple meanings which make "semantic distance" less clear, as Agent Orange is both a chemical herbicide used by military forces, an album, and a band.

Further, we observe that, on rare occasions, topic extraction followed by retrieval-augmented generation produces surprising or counterintuitive matches. In our manual review of RippleBench-Bio, this occurred in fewer than 1% of cases. For example, one question in the WMDP-bio dataset was assigned the topic Aerosol science, yet the nearest retrieved entries were *List of academic databases and search engines, List of years in science, Atmospheric science, Vector database, European Sky Shield Initiative, Document, Astroinformatics, Vaghela dynasty, HITS algorithm, and Møller scattering*. While formally correct, these associations are tangential to the specialized experimental setup described in the original question. In appendix Figure 5 we include a visualization of semantic distance for a WMDP-topic (Anthrax).

3.7 Handling duplications

Because semantic distances are computed independently for each unlearned topic, the same evaluation topic can appear at different ranks across targets. For example, unlearning topic A may retrieve $\{X, Y, Z\}$, while unlearning topic B retrieves $\{G, H, X\}$, placing X at two distinct semantic-distances. We include this knowledge-delta for both distances, assuming the model behaves consistently on the same evaluation topic regardless of context.

This averaging is a deliberate design choice: one could instead collapse duplicates to the smallest semantic-distance or weight them by occurrence, but we prioritize simplicity and comparability across methods. Duplication is practically significant in RippleBench, where overlapping neighborhoods are common.

3.8 Description of `RippleBench-Bio`

We apply `RippleBench-Maker` to the WMDP-Bio dataset, which contains $1{,}273$ unique questions on topics related to bioweapons [1], a common target domain for machine unlearning in safety research. The resulting `RippleBench-Bio` dataset spans $547{,}266$ unique topics and $352{,}961$ unique questions, distributed across semantic distances defined by WikiRAG ranks from 1 to 1000 in steps of 5. Because each unlearned target retrieves its own neighborhood, topics often appear multiple times at different distances (e.g., "vaccines" may be retrieved in relation to both "Bird Flu" and "Peptides"). In total, the dataset contains $547{,}266$ topic entries ($\sim$10% unique) and $2{,}729{,}960$ question entries.

4 Experiments

We apply the RippleBench pipeline to construct **RippleBench-WMDP-Bio** described in section 3.8, an evaluation set derived from WMDP-Bio. Our experiments measure how unlearning harmful knowledge about biological and chemical agents impacts performance on related topics at varying semantic distances.

4.1 Experimental Setup

Unlearning Methods and Model. We use Llama3-8b-Instruct [27], a fine-tuned version of Llama 3 optimized for helpful assistant behavior. We evaluate eight approaches: Gradient Difference (GradDiff) [28], Random Misdirection for Unlearning (RMU) [29], RMU with Latent Adversarial Training (RMU+LAT) [17], Representation Noising (RepNoise) [19], Erasure of Language Memory (ELM) [30], Representation Rerouting (RR) [16], Tamper Attack Resistance (TAR) [18], and PullBack & proJect (PB&J) [31]. We use publicly unlearned models, as described and shared by previous work [32]. These methods are describe in more detail in Appendix section A.1.

Evaluation. Models are evaluated on the full RippleBench datasets. `RippleBench-Bio` contains $547{,}266$ unique topics ($547{,}266$ total topic entries across all distances) and $352{,}961$ unique questions ($2{,}729{,}960$ total question entries).

4.2 Main Results: The Ripple Effect

Ripple Effects across Methods. Figure 2 compares accuracy on RippleBench-Bio across semantic distances for the base Llama3 model and several unlearning methods. As expected, the base model maintains consistently high accuracy across the full distance spectrum. All unlearning methods, by contrast, show pronounced accuracy reductions at distance 1 (the directly unlearned topics), reflecting successful suppression of targeted knowledge. The magnitude of this drop varies: methods such as GRADDIFF and TAR reduce performance by more than 25% relative to baseline, whereas others (e.g., RMU-LAT, RR, PBJ) show more moderate effects.

Beyond the immediate targets, accuracy generally recovers with increasing distance, though residual degradation remains visible even past distance 50. In relative terms, methods like RMU-LAT and RR appear to balance forgetting with less collateral impact, while approaches such as GRADDIFF and TAR emphasize stronger forgetting at the cost of wider ripple effects.

Across all unlearning methods, the checkpoint progression plots exhibit broadly similar curve shapes. In particular, we observe consistent drops and rises in accuracy at the same points along the semantic

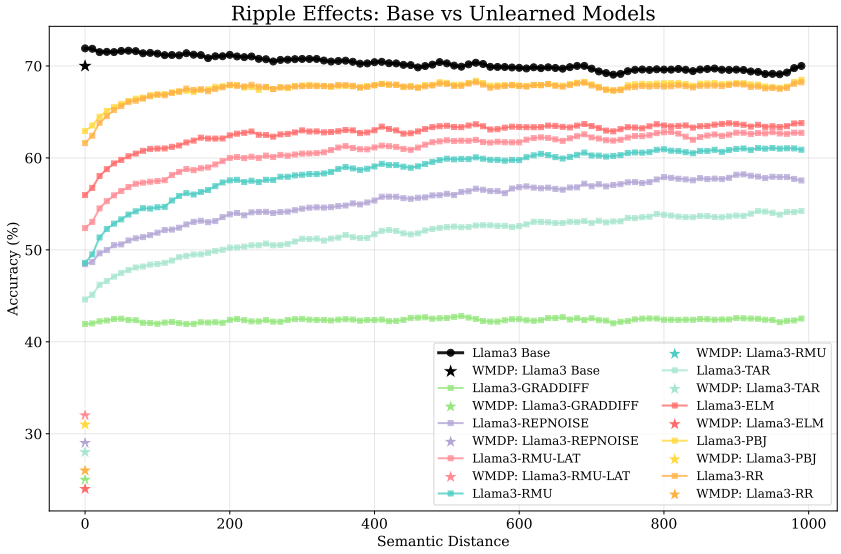

Figure 2: Ripple effects of unlearning methods on model performance across semantic distances. The base model (black) maintains consistently high accuracy, while unlearning methods show varying degrees of collateral degradation. ELM exhibits a smooth recovery with distance, whereas methods like TAR and GradDiff cause steep and persistent drops across all distances. We place stars to signify the utility of these methods on the baseline WMDP-bio dataset. To emphasize broader trends in the ripple effect, results are smoothed using a rolling average with window size 3 across distance.

distance axis. This pattern is not unexpected: since all methods share the same underlying base model and are evaluated on the same dataset, the fluctuations in accuracy reflect properties of the evaluation set itself (i.e., the same distribution of questions) rather than differences introduced by the unlearning procedure. The main distinctions between methods therefore lie not in the qualitative shape of the curves, but in their overall level of accuracy and the extent to which unlearning shifts performance relative to the baseline.

4.3 The Bomb-Next-Door: The gap in unlearning between WMDP and neighboring concept

We find a large discrepancy between reported unlearning on WMDP-Bio and performance on neighboring questions in RippleBench-WMDP-Bio. While models appear to forget the exact WMDP items (distance 0), accuracy remains much higher on distance-1 variants, revealing that unlearning is often narrowly localized to specific examples rather than the underlying concepts.

This gap likely arises from two factors: (i) current methods suppress surface forms rather than reshaping conceptual representations, and (ii) polysemanticity in language creates misleading neighbors (e.g., “mole” means one thing in chemistry and another zoology). Together, these suggest that WMDP metrics overstate forgetting. Future benchmarks should reduce polysemantic artifacts, while unlearning methods should be evaluated on their ability to generalize across semantic neighborhoods.

4.4 Evaluating the Ripple Effect over Unlearning Time

In addition to comparing unlearning methods at a single checkpoint, we also study how ripple effects evolve over the course of training. Each unlearning algorithm is checkpointed at eight stages (ckpt1–ckpt8) [7], enabling us to track the *knowledge-delta* as a function of unlearning time.

Figure 3 illustrates this progression for two representative methods, RMU and ELM. Both methods show strong suppression of accuracy at distance 0 (the directly unlearned topics), with effects that persist into nearby distances. As training progresses, the size of the ripple effect changes: in RMU, performance steadily decreases across checkpoints, while in ELM, accuracy initially drops but then partially recovers at later checkpoints, indicating a more dynamic balance between forgetting and retention.

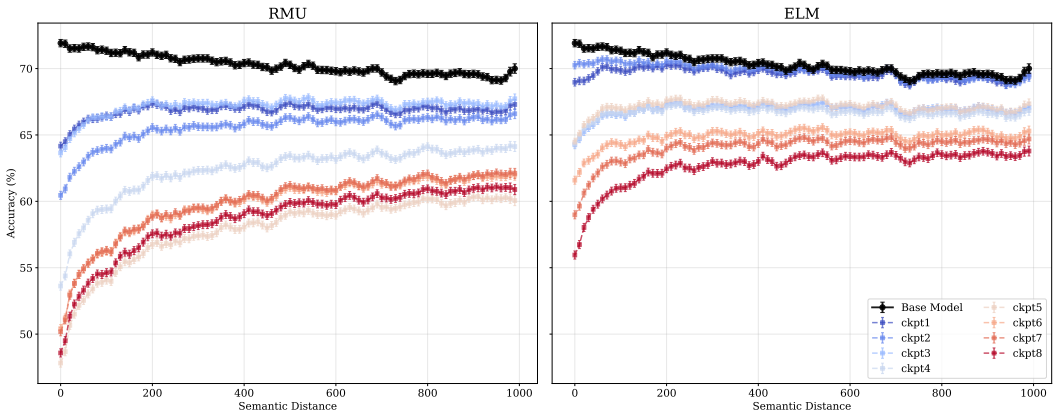

Figure 3: `RippleBench-Bio` utility over unlearning checkpoints for ELM and RMU unlearning methods.

Figure 4 extends this analysis to all methods, showing accuracy trends at three representative distances: 1, 50, and 500. Across all three settings, the curves are broadly similar—methods that induce stronger forgetting near the target also tend to produce larger ripple effects at greater distances. This consistency suggests that the overall trade-off between forgetting and collateral impact is largely stable across semantic distances.

An interesting observation is GRADDIFF, which exhibits a non-monotonic trajectory: performance declines sharply through early checkpoints but then recovers at later stages. This pattern highlights that ripple effects can evolve dynamically over training time, even when the aggregate shapes of the curves remain similar across distances. We plot this in greater detail in the Appendix, in Figure 9.

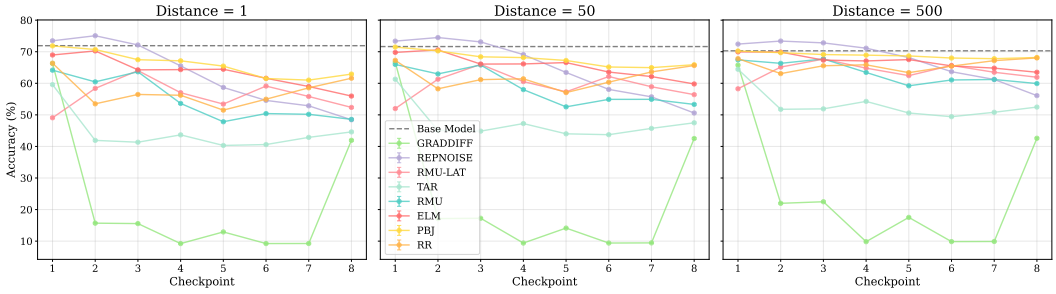

Figure 4: For 3 different semantic distances, we plot the utility over unlearning checkpoints.

5 Conclusion

We introduced `RippleBench-Maker`, a general-purpose evaluation framework, along with `RippleBench-Bio`for measuring ripple effects in machine unlearning. Our work highlights a central challenges: defining semantic distance in a way that aligns with human goals, and provides tooling and framing for designing methods that minimize collateral damage to related concepts. By providing a framework, tool, and datasets, we aim to support the development of unlearning techniques that enable precise, predictable forgetting while mitigating unintended ripple effects.

5.1 Acknowledgment of AI Use

Large language model (LLM) tools were employed to assist with data analysis and limited prose refinement. All text was originally written by the authors and carefully reviewed by human researchers.

6 Acknowledgments

Some of this work was conducted while RR was doing the ML Alignment and Theory Scholars (MATS) program and was supported by that during this time. The rest of this work, RR was supported by the NSF BCS-2218803 grant, as well as a grant from Coefficient Giving. UB is supported by a Kempner Institute Graduate Research Fellowship. This work has been made possible in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University. RG is supported by Open Philanthropy and National Science Foundation (Grant Number:NSF-2403303)

References

[1] Nathaniel Li, Alexander Patel, Elham Sidani, Maheshan Sooriyabandara, Melody Wen, Cameron Allan, Silas Watts, Shrimai Gupte, Evan Smith, Kiera Kelley, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

[2] Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models, 2023.

[3] Fazl Barez, Tingchen Fu, Ameya Prabhu, Stephen Casper, Amartya Sanyal, Adel Bibi, Aidan O'Gara, Robert Kirk, Ben Bucknall, Tim Fist, et al. Open problems in machine unlearning for ai safety. *arXiv preprint arXiv:2501.04952*, 2025.

[4] Eleni Triantafillou, Peter Kairouz, Fabian Pedregosa, Jamie Hayes, Meghdad Kurmanji, Kairan Zhao, Vincent Dumoulin, Julio Jacques Junior, Ioannis Mitliagkas, Jun Wan, Lisheng Sun Hosoya, Sergio Escalera, Gintare Karolina Dziugaite, Peter Triantafillou, and Isabelle Guyon. Are we making progress in unlearning? findings from the first neurips unlearning competition, 2024.

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

[6] Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C. Lipton, J. Zico Kolter, and Pratyush Maini. Openunlearning: Accelerating llm unlearning via unified benchmarking of methods and metrics, 2025.

[7] Zora Che, Stephen Casper, Robert Kirk, Anirudh Satheesh, Stewart Slocum, Lev E McKinney, Rohit Gandikota, Aidan Ewart, Domenic Rosati, Zichu Wu, et al. Model tampering attacks enable more rigorous evaluations of llm capabilities. *arXiv preprint arXiv:2502.05209*, 2025.

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

[9] 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.

[10] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

[11] Fan Liu, Zhao Xu, and Hao Liu. Adversarial tuning: Defending against jailbreak attacks for llms. *arXiv preprint arXiv:2406.06622*, 2024.

[12] Lei Yu, Virginie Do, Karen Hambardzumyan, and Nicola Cancedda. Robust llm safeguarding via refusal feature adversarial training. *arXiv preprint arXiv:2409.20089*, 2024.

[13] Stephen Casper, Lennart Schulze, Oam Patel, and Dylan Hadfield-Menell. Defending against unforeseen failure modes with latent adversarial training. *arXiv preprint arXiv:2403.05030*, 2024.

[14] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pages 1–14, 2025.

[15] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *URL https://arxiv. org/abs/2310.02238*, 1(2):8, 2023.

[16] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. *Advances in Neural Information Processing Systems*, 37:83345–83373, 2024.

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

[18] Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms. *URL https://arxiv. org/abs/2408.00761*, 2024.

[19] Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising effectively prevents harmful fine-tuning on llms. *CoRR*, 2024.

[20] Phillip Guo, Aaquib Syed, Abhay Sheshadri, Aidan Ewart, and Gintare Karolina Dziugaite. Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization. *arXiv preprint arXiv:2410.12949*, 2024.

[21] Stefan Schoepf, Michael Curtis Mozer, Nicole Elyse Mitchell, Alexandra Brintrup, Georgios Kaissis, Peter Kairouz, and Eleni Triantafillou. Redirection for erasing memory (rem): Towards a universal unlearning method for corrupted data. *arXiv preprint arXiv:2505.17730*, 2025.

[22] Aashiq Muhamed, Jacopo Bonato, Mona T Diab, and Virginia Smith. Saes can improve unlearning: Dynamic sparse autoencoder guardrails for precision unlearning in llms. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025.

[23] Xu Wang, Zihao Li, Benyou Wang, Yan Hu, and Difan Zou. Model unlearning via sparse autoencoder subspace guided projections. *arXiv preprint arXiv:2505.24428*, 2025.

[24] Rohit Gandikota, Sheridan Feucht, Samuel Marks, and David Bau. Erasing conceptual knowledge from language models. *arXiv preprint arXiv:2410.02760*, 2024.

[25] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

[26] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

[27] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya

Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich

Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024.

[28] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254. PMLR, 2022.

[29] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

[30] Rohit Gandikota, Sheridan Feucht, Samuel Marks, and David Bau. Erasing conceptual knowledge from language models. *arXiv preprint arXiv:2410.02760*, 2024.

[31] Anonymous. Unlearning in large language models via activation projections. 2025.

[32] Zora Che, Stephen Casper, Robert Kirk, Anirudh Satheesh, Stewart Slocum, Lev E McKinney, Rohit Gandikota, Aidan Ewart, Domenic Rosati, Zichu Wu, Zikui Cai, Bilal Chughtai, Yarin Gal, Furong Huang, and Dylan Hadfield-Menell. Model tampering attacks enable more rigorous evaluations of llm capabilities, 2025.

[33] Phillip Li, Huiwen Li, and Alexander Patel. Representation misdirection for unlearning. *arXiv preprint arXiv:2404.03233*, 2024.

[34] Lev E McKinney, Anvith Thudi, Juhan Bae, Tara Rezaei Kheirkhah, Nicolas Papernot, Sheila A McIlraith, and Roger Baker Grosse. Gauss-newton unlearning for the llm era. In *ICML 2025 Workshop on Machine Unlearning for Generative AI*, 2025.

A Supplementary Material

A.1 Existing Unlearning Techniques

The unlearning methods evaluated by Che et al. (2025) can be broadly categorized based on their underlying mechanism. Below, we briefly summarize each technique as described in their work.

Gradient and Loss-Based Fine-Tuning These methods adapt the standard fine-tuning process by modifying the loss function to de-emphasize or penalize unwanted knowledge.

- **Gradient Difference (GradDiff):** Inspired by [28], this approach trains the model to maximize the difference between the loss on the data to be forgotten and the loss on data to be retained.

- **Representation Noising (RepNoise):** Proposed by [19], this method adds a noise-inducing loss term. It encourages the model's internal representations for harmful inputs to match a simple Gaussian noise distribution.
- **Erasure of Language Memory (ELM):** Introduced by [24], ELM trains a model to mimic the behavior of an "unknowledgeable" model on the target domain, effectively erasing the specific concepts.

Representation and Activation Manipulation These techniques intervene more directly on the model's internal activations to suppress or redirect information flow related to the unwanted concepts.

- **Random Misdirection for Unlearning (RMU):** From [33], this technique involves perturbing model activations for harmful inputs while explicitly preserving activations for benign ones.
- **RMU with Latent Adversarial Training (RMU+LAT):** An extension by [17], this method strengthens RMU by using adversarial attacks in the latent space during training on the forget set.
- **Representation Rerouting (RR):** Also known as "circuit breaking" ([16]), this technique trains the model to map latent states associated with unwanted topics to orthogonal, unrelated representations.
- **K-FAC for Distribution Erasure (K-FADE):** This approach from [34] learns a set of projections in the activation space that maximally degrade performance on the forget set while minimally impacting a broader retain distribution.

Meta-Learning for Robustness This category focuses on training the model to be inherently resistant to tampering attacks.

- **Tamper Attack Resistance (TAR):** Proposed by [18], TAR is a meta-learning approach that preemptively trains a model to be robust against a fine-tuning adversary, making it harder to undo the unlearning.

A.2 Translating RAG Scores into Semantic Distance

To operationalize semantic distance, we rely on RAG rank. In this section we aim to build some intuition for how RAG ranks are constructed from underlying cosine similarity scores between Wikipedia article embeddings retrieved by Wiki-RAG. Figure 5 illustrates this process for the seed topic *Anthrax*. High-scoring neighbors such as *Anthrax weaponization* or *Bacilli* appear at low ranks, indicating close semantic proximity. As rank increases, retrieved topics gradually become less relevant (e.g., *Lobar pneumonia*) before eventually diverging to unrelated entries (e.g., *List update problem*, *List of years in politics*). This curve highlights the long tail of retrieval and motivates our bucketization of distances: low ranks capture tightly connected knowledge, while higher ranks provide semantically distant or noisy contexts.

A.3 Unlearning over time

B Progression of unlearning across checkpoints

In this appendix, we show the progression of accuracy across semantic distance for each unlearning method applied to the WMDP-bio benchmark. Each plot compares the Llama baseline to eight successive unlearning checkpoints (ckpt1–ckpt8).

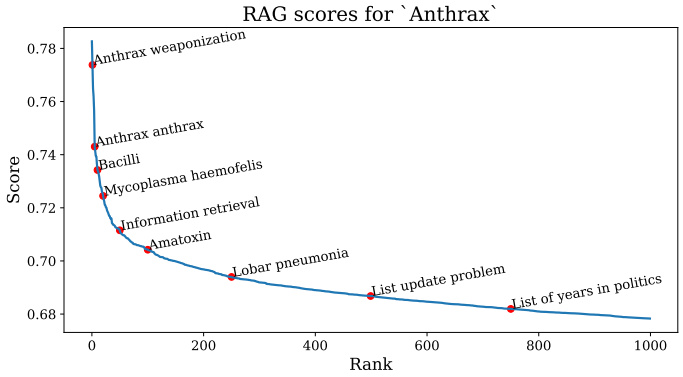

Figure 5: Example of RAG similarity scores for the seed topic *Anthrax*. Closely related neighbors (left) receive high similarity scores, while more distant or irrelevant topics (right) appear at lower scores and higher ranks. This mapping provides intuition for how semantic distance is defined and bucketized in RippleBench.

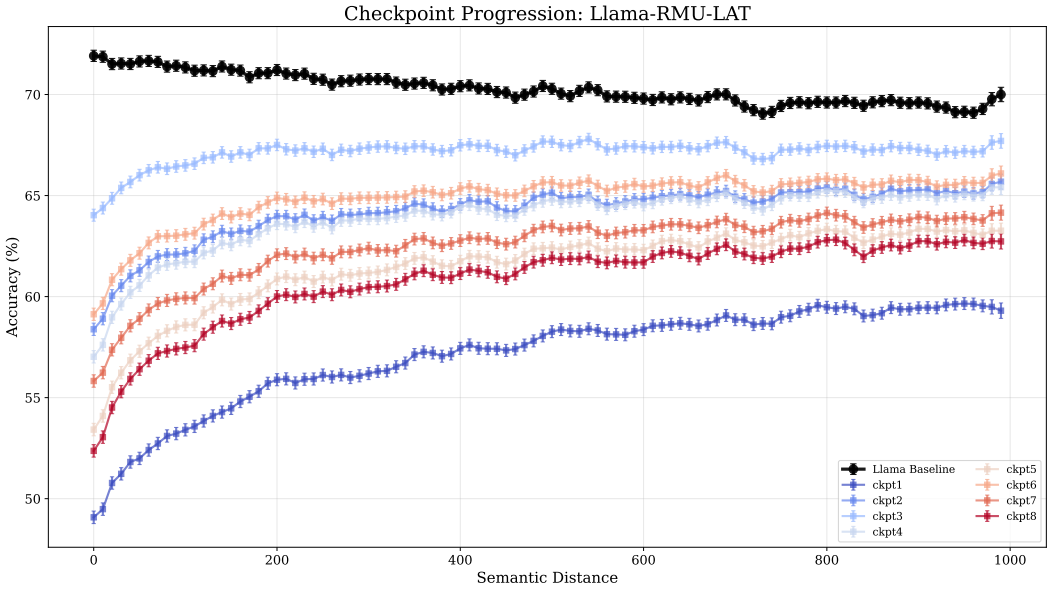

Figure 6: Checkpoint progression for **Llama-RMU-LAT**. Accuracy over semantic distance is plotted for the baseline and 8 unlearning checkpoints.

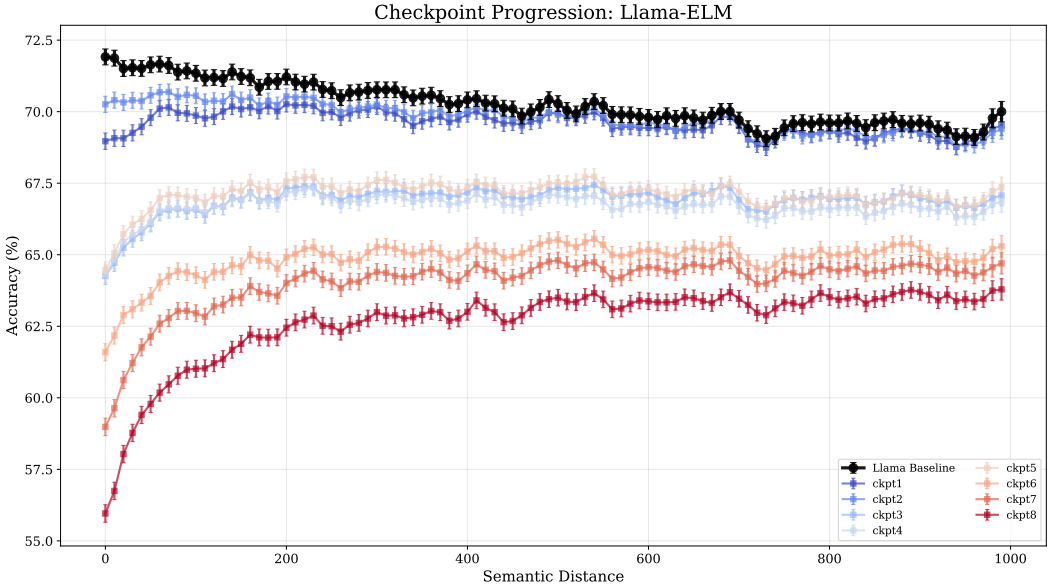

Figure 7: Checkpoint progression for **Llama-ELM**. Accuracy over semantic distance is plotted for the baseline and 8 unlearning checkpoints.

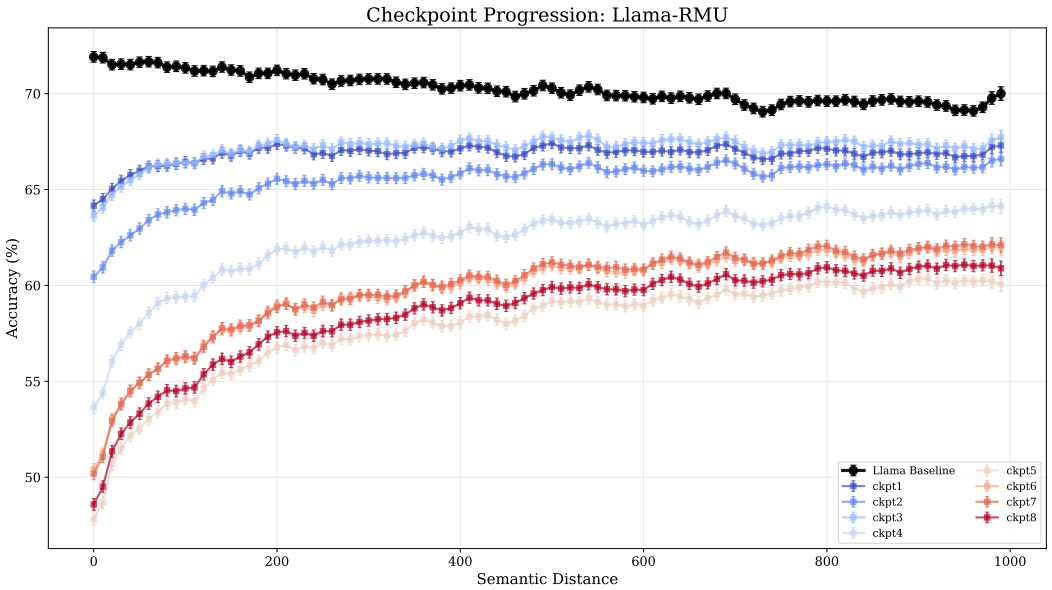

Figure 8: Checkpoint progression for **Llama-RMU**. Accuracy over semantic distance is plotted for the baseline and 8 unlearning checkpoints.

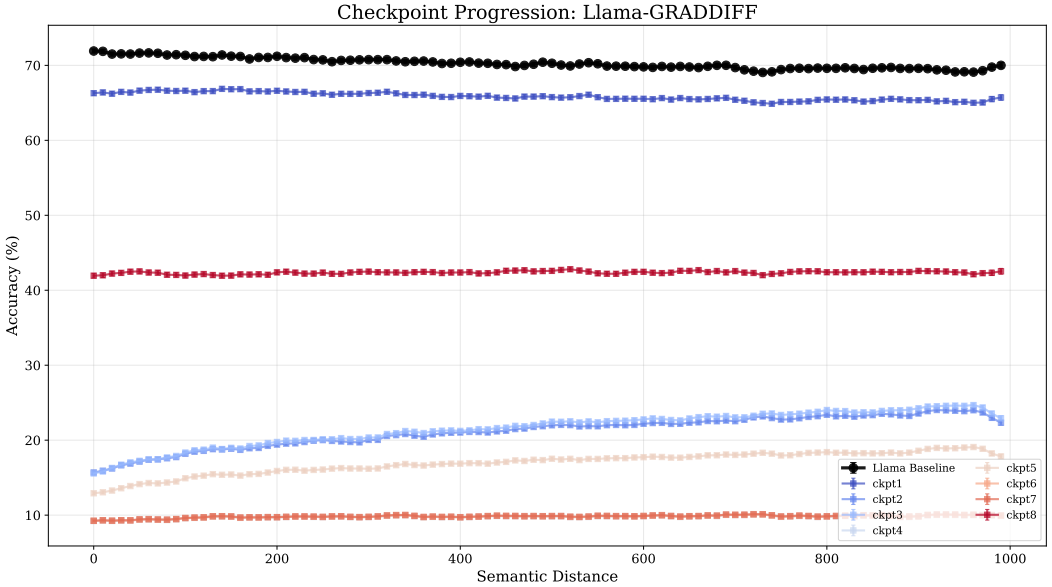

Figure 9: Checkpoint progression for **Llama-GradDiff**. Accuracy over semantic distance is plotted for the baseline and 8 unlearning checkpoints.

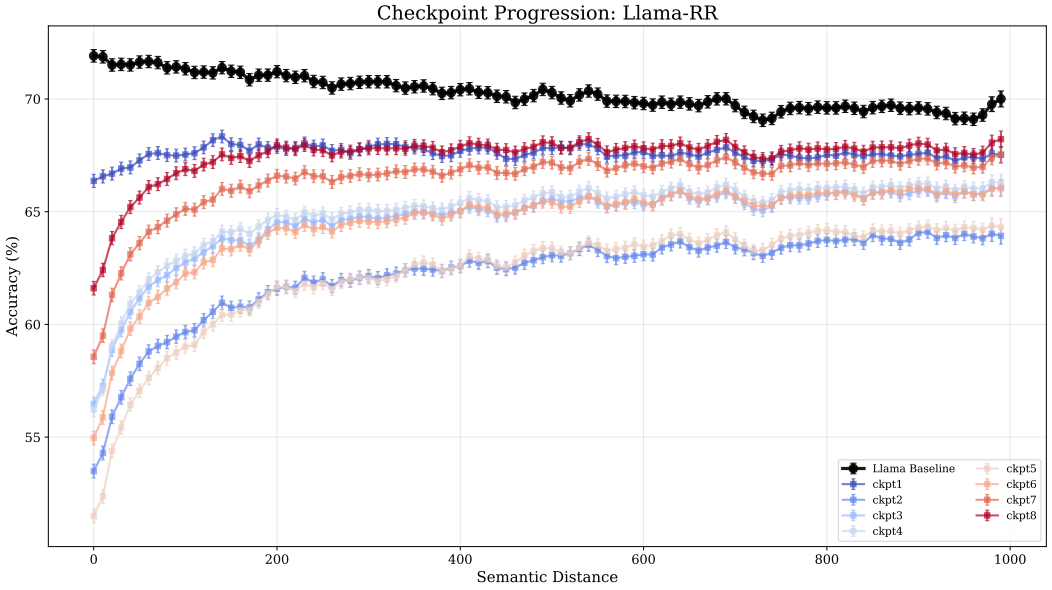

Figure 10: Checkpoint progression for **Llama-RR**. Accuracy over semantic distance is plotted for the baseline and 8 unlearning checkpoints.

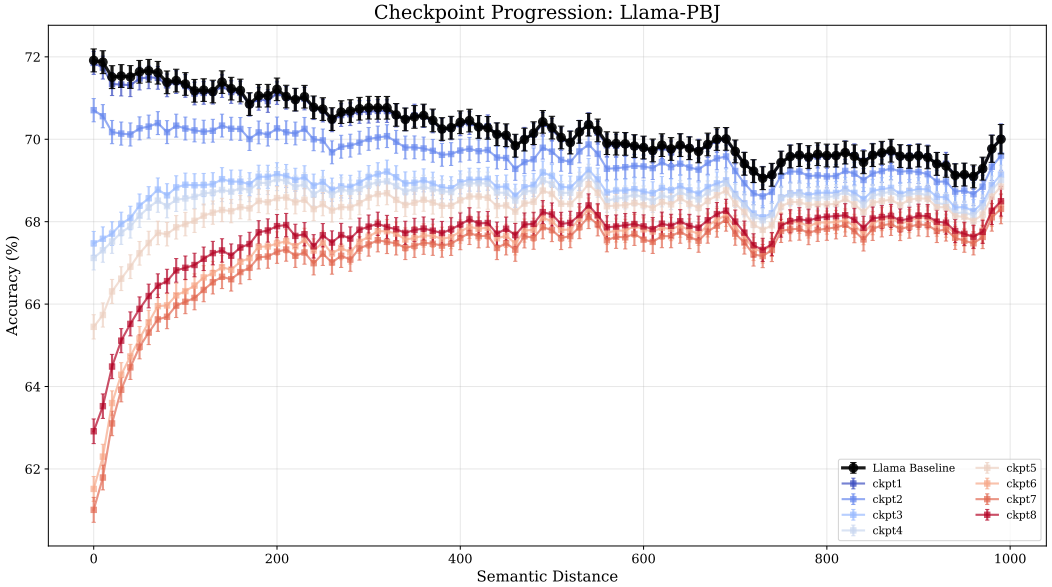

Figure 11: Checkpoint progression for **Llama-PBJ**. Accuracy over semantic distance is plotted for the baseline and 8 unlearning checkpoints.

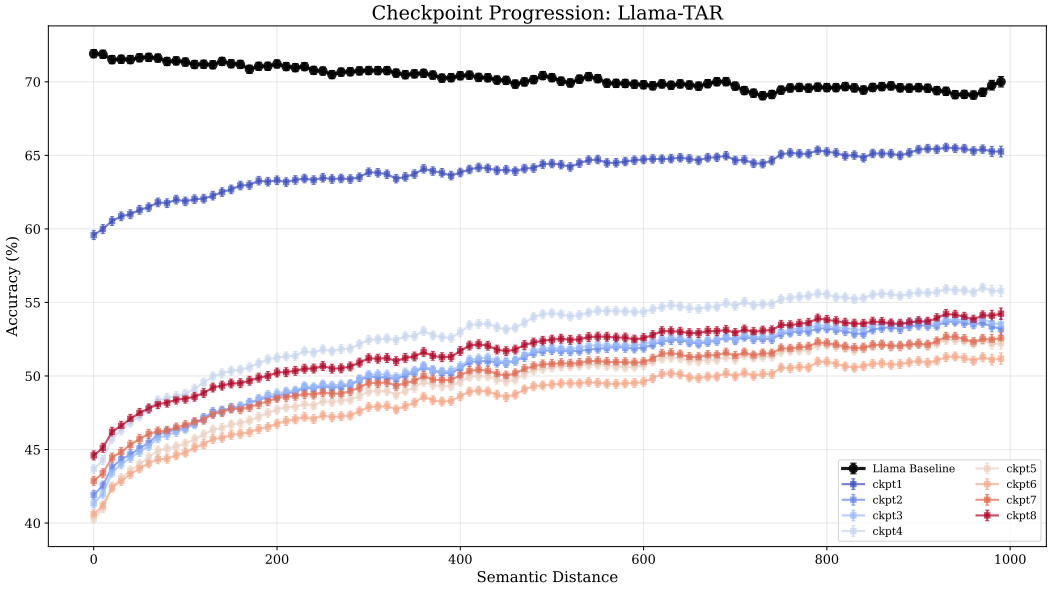

Figure 12: Checkpoint progression for **Llama-TAR**. Accuracy over semantic distance is plotted for the baseline and 8 unlearning checkpoints.

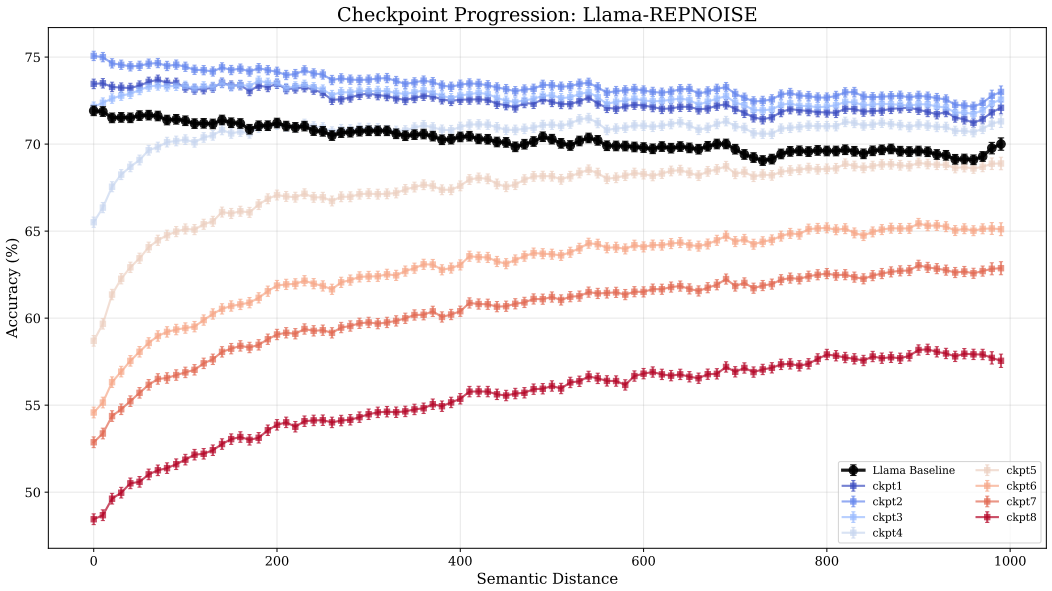

Figure 13: Checkpoint progression for **Llama-RepNoise**. Accuracy over semantic distance is plotted for the baseline and 8 unlearning checkpoints.