

REMIND: Memorization and Unlearning in LLMs Through the Lens of Input Loss Landscapes

Liran Cohen*
Technion

Yaniv Nemcovesky†
Technion

Avi Mendelson‡
Technion

Abstract

Understanding how large language models (LLMs) store, retain, and remove knowledge is critical for interpretability, reliability, and privacy compliance. We reveal a key phenomenon: machine unlearning imprints distinct geometric signatures in the model’s input loss landscape (ILL), with unlearned examples forming flat, low-curvature plateaus contrasting the sharp, high-curvature basins of retained or unseen examples, even when pointwise losses overlap, exposing residual memorization through input-output behavior alone. Building on this, we introduce REMIND (Residual Memorization in Neighborhood Dynamics), a framework that diagnoses memorization states (retained, forgotten, holdout) by probing local ILL curvature over semantically coherent neighborhoods, using only loss queries and a novel embedding-proximity perturbation method for generating controlled, interpretable variants. REMIND achieves 82% multi-class ROC-AUC in aggregate evaluations, outperforming baselines like ROUGE-L and MIN-K%+, with roughly 2x higher AUC at 1% FPR, and remains robust on paraphrased inputs. This neighborhood-level geometric analysis provides a practical, interpretable lens on LLM knowledge retention and unlearning, detecting subtle residual signals missed by pointwise or aggregated metrics.

*Email: Liranc6@gmail.com

†Email: yanemcovsky@campus.technion.ac.il

‡Email: avi.mendelson@technion.ac.il

1 Introduction

Understanding which information an LLM holds is a central challenge in NLP explainability. Machine unlearning, the task of removing the influence of specific data from a trained model, has gained importance as LLMs are deployed at scale. These models are often trained on extensive datasets containing sensitive or copyrighted content, raising legal and ethical concerns about privacy and data ownership (Geng et al., 2025). Regulations such as the General Data Protection Regulation (GDPR) (Council of European Union, 2016) enforce the "right to be forgotten" and require the removal of personal data upon request. Full model retraining can satisfy such requirements, but is typically infeasible for large models, motivating targeted unlearning methods. A critical challenge is evaluating whether such methods successfully eliminate the influence of unlearned data on model behavior.

Residual memorization refers to lingering traces of forgotten data that persist after unlearning (Hsu et al., 2025). This influence often manifests in model behavior on semantically related inputs, even when the original sample is no longer explicitly recalled. Such behavior undermines unlearning objectives by allowing indirect leakage, indicating neighborhood-level analysis is essential.

We can divide examples into three distinct memorization states: Retained (inputs from the train-

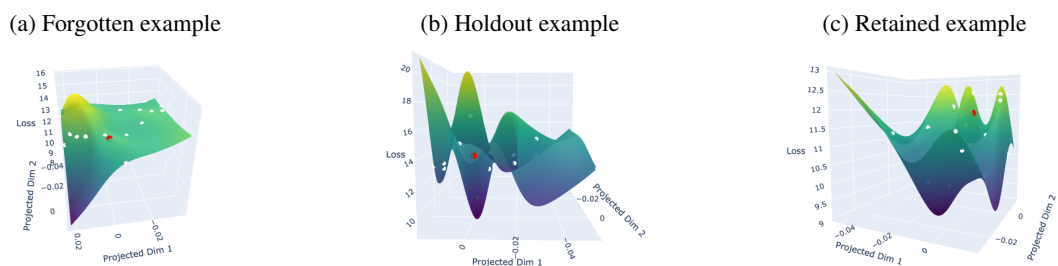


Figure 1: Input Loss Landscapes (ILL) for forgotten, holdout, and retained examples. Showing the input target (red) and its perturbed neighbors (white). Loss is visualized on a 2D embedding space via dimensionality reduction.

ing set not targeted for unlearning), Forgotten (inputs from the training set targeted for unlearning), and Holdout (unseen inputs not encountered during training or unlearning).

Input loss landscape (ILL) denotes the loss geometry around an input, obtained by evaluating the model’s loss over small perturbations within its neighborhood (§2.2). We reveal that unlearning is associated with systematic and interpretable geometric signatures in the ILL: unlearned examples form flat, low-curvature plateaus, in contrast to the sharp, high-curvature basins characteristic of trained-but-not-unlearned and unseen examples. These observations reveal distinct patterns in model behavior inferred from input-output patterns, clarifying the behavioral signatures of knowledge retention and removal. They further provide a diagnostic method for identifying residual memorization, and contribute to explainability by improving understanding of model decision processes.

Current unlearning evaluation methods in existing benchmarks (Maini et al., 2024; Shi et al., 2024; Li et al., 2024a; Dorna et al., 2024) typically examine a model’s behavior on exact forgotten sequences, and some aggregate neighbor losses using mean or max statistics (§2.1), but such pointwise or aggregated probes do not capture the geometric structure of ILL. We show that curvature-based patterns enable classifying examples into the three memorization states mentioned.

We introduce REMIND (Residual Memorization in Neighborhood Dynamics), a diagnostic evaluation framework that probes geometric signatures using only loss queries over semantically coherent input neighbors, requiring no access to model parameters or gradients. It extracts curvature-aware features from local loss landscapes and outperforms baselines on original inputs and on paraphrased variants. By revealing geometric patterns that distinguish memorization states, REMIND provides both (1) a scientific lens for understanding which information an LLM holds and how forgetting is reflected in its behavior, and (2) a practical evaluation method for diagnosing retention and unlearning, offering insight into the processes of complex NLP models.

Our approach builds on the assumption that language models exhibit local smoothness around input samples. In practice, models are trained with objectives and regularization encouraging such behavior. Consequently, memorization effects extend beyond individual training points into surrounding

neighborhoods. Thereby, variations of memorized sequences exhibit similar loss patterns, reflected in the ILL geometry. Distinct geometries, such as *shallow loss landscapes* (low variance, low gradient magnitude, low volatility), or *sharp valleys*, where loss changes abruptly (high variance, strong gradients, high volatility), can indicate memorization even when the exact sequence is not queried, giving an advantage over point-based methods. The outcome of this motivation is confirmed and demonstrated in Figure 1, which illustrates how a noisy query near a memorized point reveals informative neighborhood structure due to the model’s smooth generalization.

Our main contributions are:

- We reveal a key phenomenon: ILL geometric patterns expose residual memorization persisting after unlearning, providing insight into which information an LLM holds.
- We demonstrate that neighborhood-level geometric analysis is essential for evaluating unlearning, capturing subtle signals missed by other metrics.
- We introduce REMIND, a black-box interpretability framework leveraging ILL curvature patterns, achieving 82% multi-class ROC-AUC and outperforming prior unlearning evaluation methods across datasets, architectures, and paraphrased inputs.
- We propose embedding-proximity perturbations, a model-agnostic, semantically grounded neighbor generation technique producing human-interpretable variants for efficient, controlled exploration of local loss geometry.

2 Related Work

Understanding model behavior inferred from input-output mappings has driven the development of diverse interpretability techniques. Attention visualization methods reveal which tokens influence predictions, while gradient-based attribution approaches like integrated gradients identify input features critical to specific outputs. Probing classifiers assess whether representations encode particular linguistic properties. However, these methods focus on feature importance or representation structure rather than the geometric organization of knowledge in input space. Our work introduces

ILL analysis as a complementary explainability lens that reveals how models organize and suppress knowledge through local geometric signatures.

2.1 Unlearning Evaluation

Machine unlearning evaluation methods predominantly employ pointwise verification: measuring loss, logits, or generation quality on exact forgotten sequences (Maini et al., 2024; Shi et al., 2024; Li et al., 2024a; Jin et al., 2024). Membership inference approaches, including Zlib compression (Carlini et al., 2021), ROUGE-L matching (Lin, 2004), and MIN-K%++ (Zhang et al., 2024), detect memorization through statistical patterns in model outputs. While U-LiRA (Hayes et al., 2024) and similar per-sample techniques offer finer-grained assessment, all share a fundamental limitation: they treat models as black boxes producing isolated predictions rather than examining the underlying geometric structure of knowledge representation. White-box methods, including gradient tracing and parameter-aware probes (Hong et al., 2025), provide deeper mechanistic insights but require architecture-specific implementations and full model access, limiting their applicability to practical deployment scenarios where models are accessed only through APIs.

2.2 ILL Analysis for interpretability

ILL geometry has proven effective for explaining model behavior in computer vision, where sharp minima correlate with memorization and flat regions indicate robust generalization (Ross and Doshi-Velez, 2018; Li and Spratling, 2023; Wu et al., 2020). In NLP, (Zheng et al., 2023) observed that adversarial examples occupy sharper input loss landscape regions, suggesting geometric patterns encode vulnerability. However, ILLs remain underexplored as general-purpose explainability tools in NLP. We show that ILL curvature patterns reveal residual memorization through distinct local geometric signatures. These patterns provide a practical diagnostic signal of unlearning’s effect on knowledge retention.

2.3 Neighborhood-Based Analysis Methods

Several membership inference techniques probe input neighborhoods (Mattern et al., 2023; Galli et al., 2024; Fu et al., 2024; Mozaffari and Marathe, 2024; Xu et al., 2024), but typically aggregate neighbor information into scalar summaries (e.g., maximum or mean loss) rather than analyzing the full geometric

structure. These methods also often require white-box access or shadow model training. Perturbation techniques for interpretability include gradient-based methods like FGSM (Chacko et al., 2024), embedding noise injection (Galli et al., 2024), and MLM-based token replacement (Fu et al., 2024). However, such approaches produce semantically incoherent variations, lack distance control, or require external models. Our embedding-proximity perturbations generate semantically faithful neighbors at controlled distances, enabling systematic exploration of local loss geometry with minimal dependencies while maintaining human interpretability.

3 REMIND: A Geometric Diagnostic Framework

REMIND presents ILL analysis as a practical, fully black-box explainability method that infers model mechanisms from input-output behavior and requires no access to model parameters or activations. It’s composed of three steps (Figure 2).

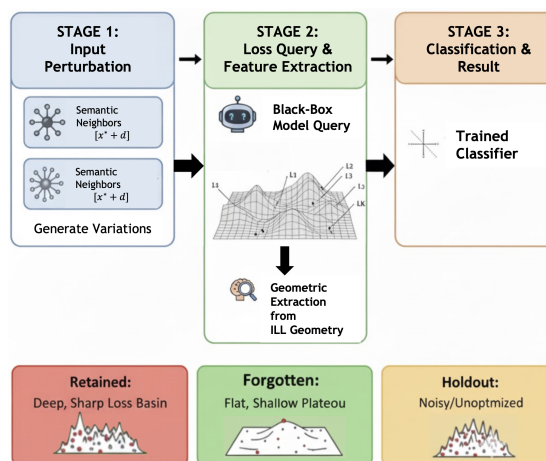


Figure 2: Methodology overview: REMIND evaluation pipeline from input to diagnostic output.

3.1 Step 1: Embedding-Proximity Perturbations

To analyze the local loss manifold, we explore the input neighborhood with semantically coherent variations that preserve meaning.

Formally, let V be a token vocabulary, let $\mathbf{x}_{1:n} = \{x_i\}_{i=1}^n \in V^n$ be a textual input sequence, let $R_p(x_i, x'_i) \in \{x_i, x'_i\}$ be a random replacement function with probability p , let $U(1, m)$ be a discrete uniform distribution over some $m \in \mathbb{N}$, let $x_i^{(j)} \in V$ be the j 'th nearest neighbor of x_i based

on the corresponding embeddings' cosine similarity, denoted as $\text{COSIM}(x_i, x'_i)$, and let K denote the number of sampled perturbed variants.

Then the neighborhood set S^K of K perturbed variants is defined as:

$$S^K = \left\{ \left\{ R_p(x_i, x_i^{(j_i)}) \right\}_{i=1}^n \mid j_i \sim U(1, m) \right\}_{\alpha=1}^K \quad (1)$$

Here, S^K is a set of semantically similar variants, preserving the original input's meaning while introducing limited semantic variation, and providing diverse coverage of the local loss topology. We use S^K to compute first-order and second-order statistics over the evaluated model loss, including mean, variance, and volatility, which form the ILL feature vector components.

The algorithm for embedding-proximity perturbation (Algorithm 1) is detailed in §A. In brief, the method replaces selected tokens with their nearest cosine-similarity neighbors over the embedding space of an external text encoder.

3.2 Step 2: Curvature-Aware Feature Extraction

Our notion of curvature is an empirical proxy derived from neighborhood loss statistics. For each input x and its neighborhood S^K , we query the model's loss $\ell(\cdot)$ and extract a feature vector of 14 geometric descriptors capturing local loss sensitivity and structure in the ILL:

feature_vector =

$$\left[\ell_{\text{orig}} \quad , \quad \mu_{\text{neigh}} \quad , \quad \ell_{\text{max}} \quad , \quad \ell_{\text{min}} \quad , \right. \\ \left. \sigma_{\text{neigh}} \quad , \quad \sigma_{\text{neigh}}^2 \quad , \quad \Delta\mu \quad , \quad \Delta_{\text{max}} \quad , \right. \\ \left. \Delta_{\text{min}} \quad , \quad \sigma_{\Delta}^2 \quad , \quad \mu_{\nabla} \quad , \quad \nabla_{\text{max}} \quad , \right. \\ \left. \sigma_{\nabla}^2 \quad , \quad \nu_{\text{neigh}} \right]$$

Where: ℓ_{orig} : original loss ; $\mu_{\text{neigh}}, \ell_{\text{max}}, \ell_{\text{min}}$: mean, max, and min neighbor losses ; $\sigma_{\text{neigh}}, \sigma_{\text{neigh}}^2$: standard deviation and variance of neighbor losses ; $\Delta\mu, \Delta_{\text{max}}, \Delta_{\text{min}}$: mean, max, and min loss deltas relative to neighbors ; σ_{Δ}^2 : variance of loss deltas ; $\mu_{\nabla}, \nabla_{\text{max}}$: mean and max gradients with respect to embeddings ; σ_{∇}^2 : variance of gradients ; ν_{neigh} : neighborhood loss volatility .

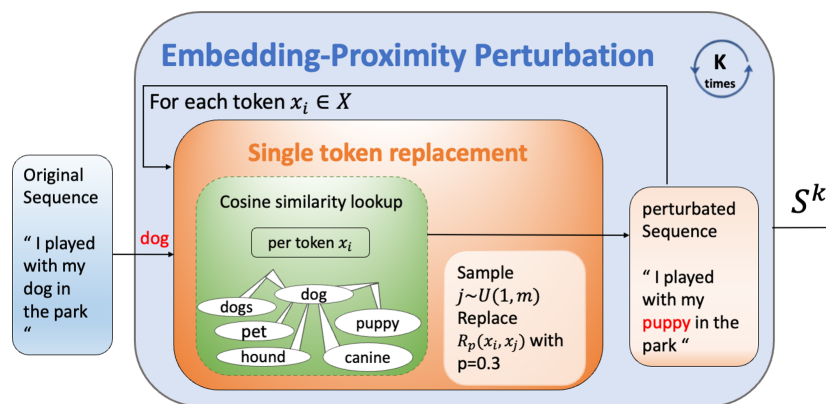


Figure 3: Illustration of the embedding-proximity perturbation algorithm.

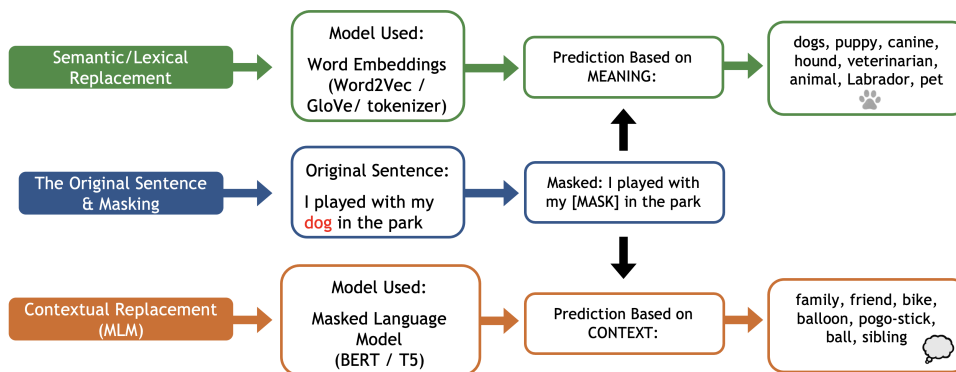


Figure 4: Comparison between MLM and embedding-proximity based perturbations illustrating differences in sampling strategies.

Note that gradient features are computed using an external text encoder. The full 14 components calculated are formalized in §A.

3.3 Step 3: Lightweight Classifier

The feature vector is passed to a lightweight classifier trained to distinguish among three classes: Retained (inputs from the training set not targeted for unlearning), Forgotten (inputs from the training set targeted for unlearning), and Holdout (unseen inputs not encountered during training or unlearning). The probe learns to recognize geometric patterns characteristic of each state.

4 Experiments: Validating Geometric Explanations

We evaluate REMIND’s ability to reveal residual memorization patterns through three research questions:

- **Q1:** *Can ILL geometric signatures reveal memorization states?*
- **Q2:** *How does REMIND compare to existing evaluation approaches?*
- **Q3:** *Do ILL-based signals remain consistently informative across various samples, architectures, and unlearning techniques?*

4.1 Experimental Setup

- Baselines: naive loss-based, ROUGE-L (Lin, 2004), Zlib Compression (Carlini et al., 2021), MIN-K% and MIN-K%++ (Zhang et al., 2024), and simplified SPV-MIA (max and mean variants) (Fu et al., 2024), adapted for unlearning evaluation.
- We evaluate on three benchmarks: MUSE (Shi et al., 2024), TOFU (Maini et al., 2024), and WMDP (Li et al., 2024a), with corresponding unlearning methods including ELM (Gandikota et al., 2024), TAR (Tamirisa et al., 2024), PB&J (McKinney et al., 2026), RMU-LAT (Li et al., 2024b), RMU (Li et al., 2024b), and SIM-NPO (Fan et al., 2024).
- We evaluate across LLMs: LLaMA-3-8B-Instruct (Grattafiori et al., 2024), LLaMA-2-7B-Chat (Touvron et al., 2023), and Zephyr-7B-Beta (Tunstall et al., 2023).

- We use the GPT-2 tokenizer for token-to-embedding mapping to ensure a model-agnostic consistency choice of neighbors selection. This ensures practical, efficient, and fair evaluation without model-specific information, reduces computation, and preserves semantic similarity across neighbors, supporting robustness even with suboptimal tokenizers.
- Primary evaluation metric: one-vs-rest ROC-AUC for the three-class task (retained/forgotten/holdout) and macro-averaged, along with ROC-AUC at 1% FPR. Also report 1vs1, F1-score, and Accuracy for completeness
- We use a labeled validation subset for calibration, consistent with prior unlearning evaluation methodologies.

Terminology: NLP benchmarks like TOFU (Maini et al., 2024) and MUSE (Shi et al., 2024) use three subsets "retain," "forget," and "holdout" to respectively represent data to keep, remove, and evaluate on; Holdout (unseen) samples are texts that were not used in training and used as a control. For the WMDP (Li et al., 2024a) benchmark, which does not include a holdout subset, we use the test set as the holdout to maintain consistency across evaluations.

4.2 Results and Discussion

RQ1: Can ILL geometric signatures reveal memorization states?

We analyze whether the ILL surrounding forgotten points exhibits clear structural patterns distinct from those of retained or holdout points (Figure 5). Notably, the ILL for forgotten points shows clear structural differences, even when input sentence loss values overlap (histogram: N). Unlearned samples show highly concentrated, low-variance distributions across nearly all metrics, including low loss variance between neighbors(D), low gradient magnitude from input sentence to its neighbors w.r.t. embeddings (A, F, I), low neighbors’ loss volatility (E), and low loss increments from input sentences to neighbors (B, G, J, L), indicating a flat, shallow loss surface. In contrast, retained and holdout samples display broader, heavy-tailed distributions

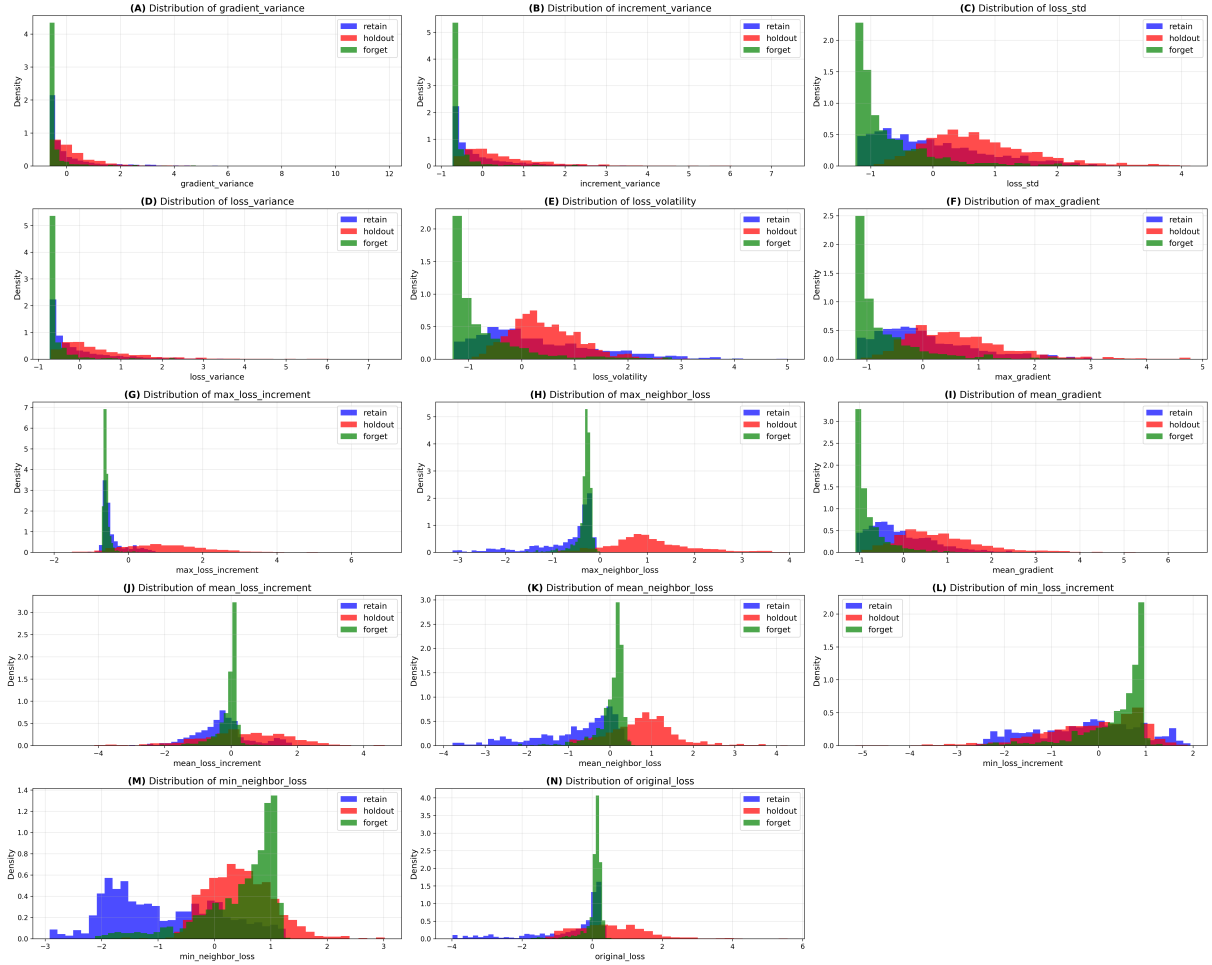


Figure 5: Density distributions of each ILL feature for retained (blue), forgotten (green), and holdout (red) examples, with x-values normalized across all examples and y-values representing density. These plots correspond to parameters: max tokens=300, test size=0.2, neighbor percentage=0.2, number of neighbors=15.

reflecting active learning dynamics and varied local curvatures. Specifically, mean, minimum, and maximum neighbor losses consistently separate the groups, with retained samples typically residing in sharp, well-fitted basins (A, B, F, G, I), holdout samples occupying more volatile or unoptimized regions (C, D, E, G, J, L), and forgotten samples lying in flat, low-sensitivity zones.

These patterns demonstrate that ILL geometric features encode distinguishable information about retained, unlearned, and holdout examples. By capturing local curvature and loss variation signatures, we provide an interpretable lens into residual knowledge, linking observed behaviors to the internal representation of unlearned information. This supports our claim that neighborhood-level geometric analysis is essential for understanding unlearning dynamics (Contribution 1 & 2).

Our observations indicate that the loss of forgotten samples is generally bounded above by the

maximum loss of holdout data and below by the minimum loss of retained examples, which may reflect the unlearning objectives to raise forgotten losses toward unseen data losses while lowering holdout loss for generalization. Forgotten inputs occupy intermediate regions of the ILL that are flatter, with reduced local variation rather than simply compressing the ILL curvature. We hypothesize that this flattening may result from the bounded loss, and leave it for future work to explore.

Q2: How does REMIND compare to existing evaluation approaches?

REMIND outperforms baselines, including output-differences-based methods, naive loss-based checks, and membership inference-style metrics, achieving higher sensitivity and lower false-negative rates in identifying unlearning failures across benchmarks (Tables 1 and 2, and Figure 6). By leveraging the full ILL geometry rather than a single aggregated signal, REMIND captures

Method	retain_vs_all_auc		forget_vs_all_auc		holdout_vs_all_auc		multi_class_auc	
	Orig	Reph	Orig	Reph	Orig	Reph	Orig	Reph
Zlib Compression (Carlini et al., 2021)	66.1871	55.39	64.3143	62.0757	21.3857	30.6443	50.63	49.37
MIN-K%++ (Zhang et al., 2024)	52.8057	45.28	51.6614	54.8843	45.5343	49.8371	50	50
ROUGE-L F1 (Lin, 2004)	63.1033	57.3317	53.8467	56.7017	66.46	60.6867	62.1733	58.355
Simplified SPV-MIA-mean (Fu et al., 2024)	47.1329	46.6243	51.1714	48.0386	51.6914	55.3357	50	50
Simplified SPV-MIA-max (Fu et al., 2024)	44.4043	44.6557	55.9643	51.5186	49.6329	53.8229	50	50
Loss based	30.0914	40.9029	48.5271	46.9657	71.3814	62.1329	50	50
Min-k% (Shi et al., 2023)	55.3386	48.9414	41.2571	49.34	53.4043	51.7229	50	50
REMIND (ours): Random Forest	82.1814	72.5029	80.6186	71.6086	85.38	78.4871	82.84	74.2571
REMIND (ours): Logistic Regression	82.3771	73.5471	77.8186	71.29	82.6514	77.9829	82.2214	75.2329

Table 1: Aggregate Comparison of Unlearning Evaluation Metrics (Part 1: ROC-AUC Metrics)

Method	retain_vs_all_auc_at_1_fp		forget_vs_all_auc_at_1_fp		holdout_vs_all_auc_at_1_fp	
	Orig	Reph	Orig	Reph	Orig	Reph
Zlib Compression (Carlini et al., 2021)	13.3029	5.28	1.38286	1.69143	0.217143	0.478571
MIN-K%++ (Zhang et al., 2024)	3.08	1.68	1.87714	2.90714	0.184286	0.4
ROUGE-L F1 (Lin, 2004)	4.84667	2.405	1.79167	2.18	2.69667	2.38833
Simplified SPV-MIA-mean (Fu et al., 2024)	0.718571	0.715714	8.72714	1.34714	9.26429	2.47714
Simplified SPV-MIA-max (Fu et al., 2024)	0.5	0.771429	7.93429	1.26143	9.19143	2.94143
Loss based	1.24857	1.72286	1.04857	0.838571	24.4	17.8886
Min-k% (Shi et al., 2023)	5.07143	2.25286	1.20429	1.57857	0.338571	0.582857
REMIND (ours): Random Forest	28.9871	14.0514	28.6229	16.5729	49.6657	39.02
REMIND (ours): Logistic Regression	26.0386	12.0371	21.9614	12.7243	43.7243	35.8529

Table 2: Aggregate Comparison of Unlearning Evaluation Metrics (Part 2: ROC-AUC at 1% FPR Metrics)

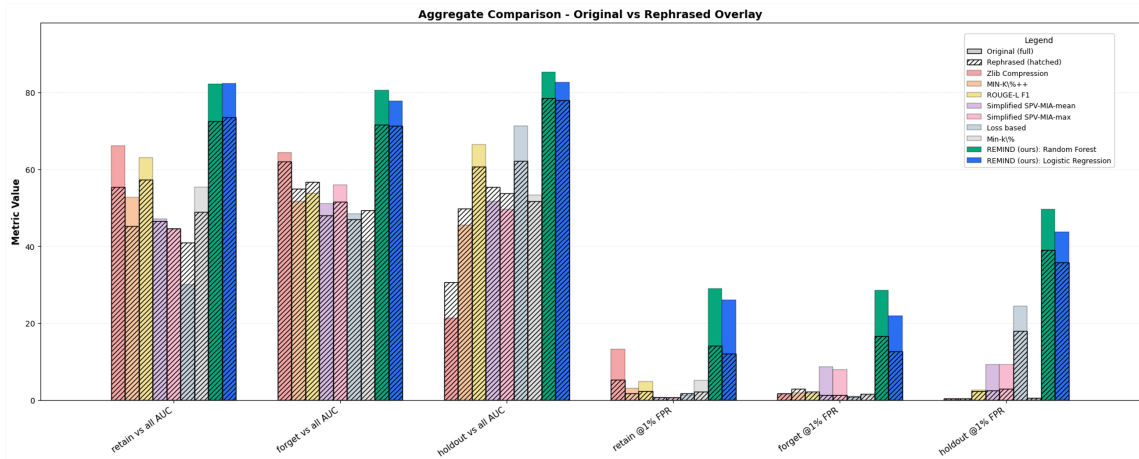


Figure 6: **Aggregate comparison of evaluation approaches.** Bar plot visualization summarizing the performance of REMIND compared to baseline metrics across different detection tasks.

subtle residual memorization patterns that other approaches miss. Its neighborhood-aware analysis systematically distinguishes retained, forgotten, and unseen examples, enabling interpretable insight into which data the model holds and robust detection of unlearning failures that are otherwise overlooked (Contribution 3).

Q3: Are ILL-based geometric features robust across inputs, models, and unlearning methods?

We evaluate REMIND across multiple axes of variation, including paraphrased inputs, different model architectures, benchmark datasets, and unlearning algorithms (Table 3).

REMIND consistently achieves high performance in all settings. These results demonstrate that the geometric patterns captured by ILL fea-

tures are stable and informative across diverse conditions, suggesting they reflect a fundamental property of how neural networks encode and suppress knowledge and supporting REMIND as a reliable, plug-in evaluation method for understanding what models retain or forget (Contributions 2 & 3).

Ablation Study REMIND is robust overall, with stable performance across the number of neighbors, moderate sensitivity to the token sampling percentage, and strongest dependence on the distance of substituted tokens. Our ablations further suggest that the ILL signatures are robust to tokenization shifts across different tokenizer architectures (bert_cased, gpt2, and t5_based), as shown in Figure 41 (see §B.3 for a detailed breakdown). We also examine transferability across un-

learning methods, models, and datasets, with detailed results reported in the Appendix (§B.4). Our analysis demonstrates strong *unlearning transferability*: classifiers trained to detect one unlearning method can successfully generalize and identify the geometric signatures of other unlearning algorithms applied to the same model and dataset. Furthermore, this transferability reveals that some unlearning methods induce broader, generalist loss geometries, while others remain highly specialized.

5 Conclusions

This paper presents a geometric perspective on residual knowledge in language models, demonstrating that unlearning fundamentally alters the Input Loss Landscape (ILL) by imprinting distinct signatures on retained, forgotten, and unseen examples. Our primary contributions include revealing how ILL curvature patterns uncover residual memorization, which has been overlooked by traditional pointwise metrics; introducing REMIND as a black-box framework for interpretability and

evaluation; and proposing embedding-proximity perturbations for generating semantically coherent neighbors. Collectively, these innovations deliver a practical diagnostic view of model behavior, surpassing baseline methods and illuminating the underlying mechanisms of knowledge retention and removal in large language models.

Geometric Perspective on Knowledge

We introduce a geometric perspective on residual knowledge in large language models, revealing that unlearning reshapes the Input Loss Landscape (ILL) rather than merely altering pointwise losses. We reveal a fundamental geometric phenomenon: forgotten inputs occupy intermediate regions of the ILL, characterized by flatter, lower variance, and less volatile neighborhoods than retained and holdout examples, while their loss values remain between them. These findings indicate that unlearning produces structured geometric signatures rather than simply increasing loss values, demonstrating that forgetting manifests as smooth but non-featureless patterns in the loss landscape. This

Model	Benchmark	Method	Classifier	Retain vs All	Forget vs All	Holdout vs All	Retain vs Forget	Retain vs Holdout	Forget vs Holdout	Multi-Class AUC	Retain vs All @ 1FP	Forget vs All @ 1FP	Holdout vs All @ 1FP	Accuracy
llama-3-8b	WMDP	ELM	LogReg	99.6	98.2	100.0	99.5	100.0	100.0	99.6	94.5	85.0	100.0	100.0
llama-3-8b	WMDP	ELM	RF	99.8	98.7	100.0	99.3	100.0	100.0	99.8	94.5	96.5	100.0	100.0
llama-3-8b	WMDP	PBJ	LogReg	93.9	75.5	87.2	88.3	99.5	82.4	85.9	31.0	6.5	5.0	78.8
llama-3-8b	WMDP	PBJ	RF	93.8	89.5	98.8	87.8	99.7	97.9	94.5	28.0	34.5	73.5	95.5
llama-3-8b	WMDP	RMU	LogReg	95.6	92.4	99.5	92.7	99.8	99.4	97.1	51.0	50.5	99.0	99.2
llama-3-8b	WMDP	RMU	RF	96.2	95.9	99.9	92.6	99.7	99.9	97.1	60.0	45.5	99.0	99.3
llama-3-8b	WMDP	RMU-LAT	LogReg	94.8	88.9	100.0	91.1	100.0	100.0	97.0	42.0	34.0	100.0	99.8
llama-3-8b	WMDP	RMU-LAT	RF	96.3	96.0	100.0	92.7	100.0	100.0	97.4	39.0	60.5	100.0	100.0
llama-3-8b	WMDP	TAR	LogReg	95.8	74.7	77.8	95.0	98.1	70.7	84.0	52.5	8.0	4.5	68.0
llama-3-8b	WMDP	TAR	RF	94.9	85.3	95.2	94.1	98.3	92.1	91.8	49.0	29.0	39.5	90.3
llama-2-7b	MUSE - News	SimNPO	LogReg	87.5	68.1	96.0	80.1	97.9	95.1	88.1	16.3	0.6	23.6	90.8
llama-2-7b	MUSE - News	SimNPO	RF	86.3	82.3	96.3	79.7	98.2	93.8	88.5	30.2	7.1	17.8	90.6
llama-2-7b	TOFU	SimNPO	LogReg	58.6	56.3	59.6	59.3	63.3	47.8	59.8	0.0	6.3	2.5	67.1
llama-2-7b	TOFU	SimNPO	RF	51.1	49.8	54.6	50.9	52.7	45.9	54.8	0.0	6.3	5.0	63.8
llama-2-7b	TOFU	SimNPO	LogReg	74.1	80.3	60.1	81.7	67.4	79.9	73.6	5.0	25.0	1.3	62.9
llama-2-7b	TOFU	SimNPO	RF	69.8	78.4	64.3	80.5	65.3	78.8	72.3	6.3	23.8	1.3	66.7
zephyr-7b	WMDP	SimNPO	LogReg	80.8	83.7	99.8	65.2	99.6	100.0	88.0	5.0	11.5	98.5	98.8
zephyr-7b	WMDP	SimNPO	RF	81.7	83.7	100.0	61.7	100.0	100.0	87.3	3.5	12.0	99.0	99.0

Table 3: Detailed scores for individual models - Original Inputs. The results are based on the following parameters: $m=20$, $K_neighbors=15$, replacement probability $p=0.6$, test size=0.2. LogReg denotes Logistic Regression, and RF denotes Random Forest classifiers. For clarity in this large table, we have not inserted citations; the citations can be found in the related work and the experimental setup.

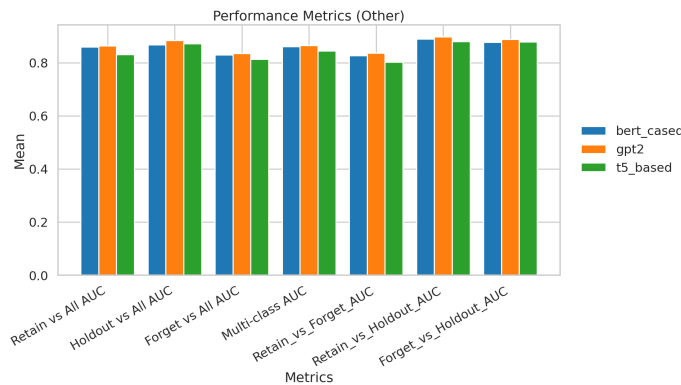


Figure 7: **Tokenizer robustness across broad metrics.**

geometry provides a new lens for NLP explainability, treating ILL geometry as a window into internal decision-making processes.

REMIND: novel interpretability framework REMIND provides a novel framework for characterizing residual knowledge in post-unlearning models. By using structured, semantically grounded perturbations to probe local geometric neighborhoods, it reveals subtle memorization signatures that remain unnoticed by standard evaluation metrics. Its black-box, model-agnostic design supports practical and human-readable diagnostic signals across paraphrases, architectures, and unlearning methods. REMIND outperforms existing black-box unlearning evaluation techniques, uncovering residual memorization that pointwise loss metrics and neighborhood aggregation methods fail to detect, highlighting its effectiveness in revealing hidden knowledge retention.

Embedding-Proximity Perturbations as a novel neighbor generation technique

Our embedding-proximity perturbation technique enables efficient exploration of semantically coherent neighborhoods, producing human-interpretable neighbor inputs. This approach creates variants that a real user might plausibly provide while avoiding the uninterpretable artifacts caused by continuous embedding noise. Compared to masked language models-based perturbations, our embedding-proximity method offers tokenizer-agnostic, semantically grounded, and computationally efficient neighbor generation that enforces semantic-distance control, avoids conditional-generation bias, and yields controlled, interpretable neighborhoods for analysis.

Advantages and Broader Implications

ILL geometry provides a principled lens to monitor unlearning dynamics and support human-in-the-loop refinement. We highlight REMIND’s potential to monitor unlearning progress during training by tracking convergence in the ILL feature space, while also noting a potential limitation and trade-off in the unlearning process. Overall, by reframing forgetting as a geometric phenomenon, we provide an interpretable, diagnostic framework for explaining what LLMs retain and forget, offering insight into observable retention patterns and how LLMs encode and remove knowledge.

Reproducibility: Complete implementation and evaluation code available at <https://anonymous.>

[4open.science/r/Input-Loss-Landscapes-ILL-Reveal-Residual-Memorization-in-Post-Unlearning-LLMs-3FC0/](https://open.science/r/Input-Loss-Landscapes-ILL-Reveal-Residual-Memorization-in-Post-Unlearning-LLMs-3FC0/)

6 Limitations

Although REMIND provides a strong black-box diagnostic for residual memorization through input loss landscape geometry, several limitations remain.

Computational overhead Our approach relies on querying the model over multiple semantically coherent perturbations for each input. This incurs a higher computational cost than pointwise metrics (e.g., a single forward pass for loss or ROUGE-L).

Tokenizer dependence The embedding-proximity perturbation method uses a fixed tokenizer (GPT-2 in our experiments) for neighbor selection. This choice, while model-agnostic and efficient, may introduce tokenizer-specific biases in semantic similarity and neighbor quality.

Fixed-K sampling Moreover, while our current fixed-K sampling methodology is efficient, future iterations could be improved via adaptive early-stopping based on loss-variance thresholds, potentially reducing the number of forward passes required for samples with highly uniform or highly erratic local geometries.

Supervised calibration requirement Our evaluation is conducted on established unlearning benchmarks (TOFU, MUSE, WMDP) with standard unlearning techniques. Our method requires classifier calibration over a small subset of labeled data (retain/forget/holdout), which may not be available in all scenarios. However, we show that the resulting classifier generalizes well across models, datasets, and unlearning methods.

Integration with unlearning methods Finally, while REMIND excels at distinguishing retained, forgotten, and holdout states, this work does not integrate these insights to improve existing unlearning algorithms.

References

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. [Extracting training data from large language models](#). In

- 30th USENIX Security Symposium (USENIX Security 21). USENIX Association.
- Samuel Jacob Chacko, Sajib Biswas, Chashi Mahiul Islam, Fatema Tabassum Liza, and Xiuwen Liu. 2024. [Adversarial attacks on large language models using regularized relaxation](#). *arXiv preprint arXiv:2410.19160*.
- Council of European Union. 2016. [Council regulation \(eu\) no 2016/679](#). General Data Protection Regulation.
- Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C. Lipton, J. Zico Kolter, and Pratyush Maini. 2024. [Openunlearning: Accelerating llm unlearning via unified benchmarking of methods and metrics](#). *arXiv preprint arXiv:2506.12618*.
- Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. 2024. [Simplicity prevails: Rethinking negative preference optimization for llm unlearning](#). *arXiv preprint arXiv:2410.07163*.
- Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2024. [Membership inference attacks against fine-tuned large language models via self-prompt calibration](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 134981–135010. Curran Associates, Inc.
- Filippo Galli, Luca Melis, and Tommaso Cucinotta. 2024. [Noisy neighbors: Efficient membership inference attacks against llms](#). *arXiv preprint arXiv:2406.16565*.
- Rohit Gandikota, Sheridan Feucht, Samuel Marks, and David Bau. 2024. [Erasing conceptual knowledge from language models](#). *arXiv preprint arXiv:2410.02760*.
- Jiahui Geng, Qing Li, Herbert Woisetschlaeger, Zongxiong Chen, Fengyu Cai, Yuxia Wang, Preslav Nakov, Hans-Arno Jacobsen, and Fakhri Karray. 2025. [A comprehensive survey of machine unlearning techniques for large language models](#). *arXiv preprint arXiv:2503.01854*.
- Arthur Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhishek Pandey, Abhishek Kadian, Ahmad Al-Dahle, Anton Letman, Aran Mathur, Beren Schelten, Blake Vaughan, and 1 others. 2024. [Llama 3: The reference implementation](#). *arXiv preprint arXiv:2407.21783*.
- Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. 2024. [Inexact unlearning needs more careful evaluations to avoid a false sense of privacy](#). *arXiv preprint arXiv:2403.01218*.
- Yihuai Hong, Lei Yu, Haiqin Yang, Shauli Ravfogel, and Mor Geva. 2025. [Intrinsic evaluation of unlearning using parametric knowledge traces](#). *arXiv preprint arXiv:2406.11614*.
- Hsiang Hsu, Pradeep Niroula, Zichang He, and Chun-Fu Chen. 2025. [Are we really unlearning? the presence of residual knowledge in machine unlearning](#). In *I Can't Believe It's Not Better: Challenges in Applied Deep Learning*.
- Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. 2024. [Rwku: Benchmarking real-world knowledge unlearning for large language models](#). *arXiv preprint arXiv:2406.10890*.
- Lin Li and Michael Spratling. 2023. [Understanding and combating robust overfitting via input loss landscape analysis and regularization](#). *Pattern Recognition*, 136:109229.
- 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, and 38 others. 2024a. [The wmdp benchmark: Measuring and reducing malicious use with unlearning](#). *arXiv preprint arXiv:2403.03218*.
- Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, and 1 others. 2024b. [The wmdp benchmark: Measuring and reducing malicious use with unlearning](#). *arXiv preprint arXiv:2403.03218*.
- Chin-Yew Lin. 2004. [Rouge: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024. [Tofu: A task of fictitious unlearning for llms](#). *arXiv preprint arXiv:2401.06121*.
- Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. [Membership inference attacks against language models via neighbourhood comparison](#). *arXiv preprint arXiv:2305.18462*.
- L. McKinney, A. Thudi, J. Bae, T. Rezaei, N. Papernot, S.A. McIlraith, and R. Grosse. 2026. [Gaussian unlearning for the llm era](#). *arXiv preprint arXiv:2602.10568*.
- Hamid Mozaffari and Virendra J Marathe. 2024. [Semantic membership inference attack against large language models](#). *arXiv preprint arXiv:2406.10218*.
- Andrew Ross and Finale Doshi-Velez. 2018. [Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

- Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. [Detecting pretraining data from large language models](#). *arXiv preprint arXiv:2310.16789*.
- Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2024. [Muse: Machine unlearning six-way evaluation for language models](#). *arXiv preprint arXiv:2407.06460*.
- Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, and 1 others. 2024. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Yasmine Almahairi, Nikolay Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, Lukas Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *arXiv preprint arXiv:2307.09288*.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourier, Nathan Habib, and 1 others. 2023. [Zephyr: Direct distillation of lm alignment](#). *arXiv preprint arXiv:2310.16944*.
- Dongxian Wu, Shu-Tao Xia, and Yisen Wang. 2020. Adversarial weight perturbation helps robust generalization. 33:2958–2969.
- Huan Xu, Zhanhao Zhang, Xiaodong Yu, Yingbo Wu, Zhiyong Zha, Bo Xu, Wenfeng Xu, Menglan Hu, and Kai Peng. 2024. Targeted training data extraction—neighborhood comparison-based membership inference attacks in large language models. *Applied Sciences*, 14(16):7118.
- Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2024. [Min-k%++: Improved baseline for detecting pre-training data from large language models](#). *arXiv preprint arXiv:2404.02936*.
- Rui Zheng, Shihan Dou, Yuhao Zhou, Qin Liu, Tao Gui, Qi Zhang, Zhongyu Wei, Xuan-Jing Huang, and Menghan Zhang. 2023. Detecting adversarial samples through sharpness of loss landscape. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11282–11298.

A Formal Definitions and Algorithms

A.1 Embedding-Proximity Perturbation Algorithm

The exact embedding-proximity perturbation technique generating these sets is detailed in Algorithm 1.

Algorithm 1 Embedding-Proximity Perturbation

Require: Input sequence $\mathbf{x} = \{x_1, \dots, x_n\}$, perturbation probability p , number of neighbors for each token m , number of perturbations K

Ensure: Set of perturbed variants $S^K = \{\tilde{\mathbf{x}}_j\}_{j=1}^K$

```

1: for  $k = 1$  to  $K$  do
2:   for  $i = 1$  to  $n$  do
3:     Sample  $j \sim U(1, m)$ 
4:     Let  $x_i^{(j)}$  be the  $j$ th nearest neighbor of  $x_i$ 
       by cosine similarity
5:     Replace  $x_i$  with  $x_i^{(j)}$  with probability  $p$ 
6:   end for
7:   Store perturbed sequence
    $\tilde{\mathbf{x}}_k = \{x'_1, \dots, x'_n\}$ 
8: end for
9: return  $S^K = \{\tilde{\mathbf{x}}_j\}_{j=1}^K$ 

```

A.2 Curvature-Aware Feature Extraction

For each input x and its neighborhood S^K , we query the model’s loss $\ell(\cdot)$ and extract a feature vector of 14 geometric descriptors capturing local loss sensitivity and structure in the ILL:

feature_vector =

$$[\ell_{\text{orig}} \quad , \quad \mu_{\text{neigh}} \quad , \quad \ell_{\text{max}} \quad , \quad \ell_{\text{min}}, \\ \sigma_{\text{neigh}} \quad , \quad \sigma_{\text{neigh}}^2 \quad , \quad \Delta\mu \quad , \quad \Delta_{\text{max}}, \\ \Delta_{\text{min}} \quad , \quad \sigma_{\Delta}^2 \quad , \quad \mu_{\nabla} \quad , \quad \nabla_{\text{max}}, \\ \sigma_{\nabla}^2 \quad , \quad \nu_{\text{neigh}}]$$

Where: ℓ_{orig} : original loss ; $\mu_{\text{neigh}}, \ell_{\text{max}}, \ell_{\text{min}}$: mean, max, and min neighbor losses ; $\sigma_{\text{neigh}}, \sigma_{\text{neigh}}^2$: standard deviation and variance of neighbor losses ; $\Delta\mu, \Delta_{\text{max}}, \Delta_{\text{min}}$: mean, max, and min loss deltas relative to neighbors ; σ_{Δ}^2 : variance of loss deltas ; $\mu_{\nabla}, \nabla_{\text{max}}$: mean and max gradients with respect to embeddings ; σ_{∇}^2 : variance of gradients ; ν_{neigh} : neighborhood loss volatility. Note that gradient features are computed using an external text encoder.

A.2.1 Formalizing Mathematical Notation

We provide our formalization.

- V – token vocabulary
- $x = (x_1, \dots, x_n)$ – input sequence of tokens
- M – target LLM
- $\ell_M(x)$ – model loss function
- $N(x)$ – semantic neighborhood of x , agnostic to the model target
- $S^K \subseteq N(x)$ – a set of semantically proximate neighbors generated via our EPP method.

We define the ILL around input x as

$$\mathcal{G}_x = \{(x', \ell_M(x')) \mid x' \in \mathcal{N}_m(x)\} \subset \mathbb{R}^{d+1} \quad (2)$$

with d being the embedding dimensionality.

The feature vector is defined as follows.

- $\ell_0 := \ell_M(x)$
- $\ell_i := \ell_M(x'_i), \quad x'_i \in S^K \subseteq N(x)$
- $\Delta_i := \ell_i - \ell_0$

Note that gradient features are computed using an external text encoder.

B Ablation Studies

We conduct ablation studies to validate design choices and assess robustness to hyperparameter variations. These experiments examine sensitivity to neighborhood construction parameters (K, m, p) and evaluate our perturbation method EPP against MLM baseline perturbation method.

B.1 Sensitivity to Neighborhood Construction Parameters

Experimental Setup. We systematically evaluate each parameter while keeping others fixed. Parameter values were initially selected arbitrarily; when abnormal behavior was identified, we added additional values around those regions for finer-grained analysis:

$$K \in \{1, 2, 5, 10, 15, 20, 30, 40\},$$

$$m \in \{2, 5, 10, 12, 14, 17, 18, 19, 20, 21, 22, 23, 24, \\ 25, 26, 27, 28, 29, 30, 35, 40, 50, 60, 70\},$$

$p \in \{0.1, 0.2, 0.3, 0.4, 0.45, 0.5, 0.55, 0.6, 0.7\}$.

For each configuration, we report multi-class ROC-AUC, one-vs-rest, and one-vs-one AUC scores of retain, forget, and holdout labels, aggregated across models, datasets, and unlearning methods. This analysis is designed to assess robustness to neighborhood construction rather than to optimize hyperparameters.

B.1.1 Perturbation Probability (p)

The parameter p controls the fraction of tokens modified when generating semantic variants, trading local fidelity for broader neighborhood exploration.

Key Findings: Figures 10, 24, 27, and 25 illustrate the consistent performance across diverse perturbation strengths.

- Classification performance remains stable and high over a broad range of perturbation probabilities. No single p value consistently dominates across tasks, reinforcing REMIND’s robustness.
- Moderate perturbation values ($p \in [0.3, 0.6]$) yield stronger performance. Interestingly, even lower or higher perturbation levels often yield similar results, indicating that REMIND does not rely on a finely tuned perturbation strength.
- Forget vs. all classification (Figure 24) is stable across perturbation probabilities, often

achieving high AUC. This suggests that unlearned examples induce distinct loss landscape geometry.

- Retain vs. forget (Figure 27) and forget vs. holdout (Figure 25) discriminations shows mild sensitivity to p , but performance remains well above chance for all perturbation levels. This indicates that the geometric distinction persistent across perturbation magnitudes.

Implications: As demonstrated in the figures, the stability of performance across a broad range of p values indicates that REMIND’s geometric characterization of memorization states is not contingent on the exact perturbation strength. This robustness reduces the need for hyperparameter tuning, allowing practitioners to adapt neighborhood construction to specific computational or domain constraints without sacrificing interpretability or classification performance.

B.1.2 Embedding-Space Neighbor Range (m)

The parameter m defines the semantic radius by controlling how many top embedding-space neighbors are considered for token replacement, with smaller values yielding conservative substitutions and larger values enabling more diverse perturbations.

Key Findings: As visualized in Figure 9, performance remains robust across the full spectrum of semantic radii.

Feature	Symbol	Definition
Original loss	ℓ_0	$\ell_M(x)$
Mean neighbor loss	μ_ℓ	$\frac{1}{K} \sum_{i=1}^K \ell_i$
Max neighbor loss	ℓ_{\max}	$\max_{1 \leq i \leq K} \ell_i$
Min neighbor loss	ℓ_{\min}	$\min_{1 \leq i \leq K} \ell_i$
Standard deviation	σ_ℓ	$\sqrt{\frac{1}{K} \sum_{i=1}^K (\ell_i - \mu_\ell)^2}$
Variance (sharpness proxy)	Var_ℓ	$\frac{1}{K} \sum_{i=1}^K (\ell_i - \mu_\ell)^2$
Mean delta	μ_Δ	$\frac{1}{K} \sum_{i=1}^K \Delta_i$
Max delta	Δ_{\max}	$\max_{1 \leq i \leq K} \Delta_i$
Min delta	Δ_{\min}	$\min_{1 \leq i \leq K} \Delta_i$
Delta variance	Var_Δ	$\frac{1}{K} \sum_{i=1}^K (\Delta_i - \mu_\Delta)^2$
Mean gradient magnitude	μ_∇	$\frac{1}{K} \sum_{i=1}^K \ \nabla_x \ell_M(x'_i)\ _2$
Max gradient magnitude	∇_{\max}	$\max_{1 \leq i \leq K} \ \nabla_x \ell_M(x'_i)\ _2$
Gradient variance	Var_∇	$\frac{1}{K} \sum_{i=1}^K (\ \nabla_x \ell_M(x'_i)\ _2 - \mu_\nabla)^2$
Neighborhood volatility	ν_x	$\sqrt{\text{Var}_\Delta}$

Table 4: Curvature-Aware Feature Vector Components

- AUC results show some sensitivity to m , with no universally optimal value identified, indicating that examples exhibit distinct loss landscape geometry even when the neighborhood spans broad semantic distances.
- Across models and metrics, values around $m \in [10, 30]$ often yield higher performance and higher variability, though performance degradation outside this range remains modest. This suggests a “sweet spot” for semantic radius that balances local coherence with sufficient neighborhood diversity, though performance degradation outside this range remains modest.
- Small neighborhoods are sufficient to recover strong geometric signals. Even for $m \leq 5$, performance remains high, indicating that memorization and unlearning signatures can be detected using small semantic neighborhoods, though they are not restricted to them.

B.1.3 Number of Sampled Variants (K)

The parameter K controls the number of perturbed variants sampled per input.

Key Findings: Figures 8 and 18 demonstrate the stability of classification performance across most K values.

- AUC results show strong performance across a wide range of K values. No single value dominates across tasks or architectures, highlighting the robust generalizability of REMIND.
- AUC scores remain high, though suboptimal, even for very small sample counts ($K = 2$), suggesting that discriminative signals can be recovered from a few perturbed variants.
- For certain tasks, most notably forget vs. hold-out classification (Figure 18), peak performance is attained at larger K , suggesting that these distinctions benefit from broader neighborhood sampling to capture more subtle geometric differences.

Implications: These results indicate that REMIND does not heavily rely on densely sampled neighborhoods to recover meaningful loss landscape structure. Strong performance even at small K values shows that informative geometric signals are present with minimal neighborhood sampling, while improvements at larger K for certain tasks

suggest that aggregating additional variants can help resolve more subtle distinctions. This flexibility enables efficient deployment with fewer loss queries while maintaining high interpretability and detection accuracy.

B.2 Embedding-Proximity Perturbation vs. Context-Based Methods

To validate our choice of Embedding-Proximity Perturbation (EPP) over context-based masked language model (MLM) approaches (e.g., BERT), we compare semantic preservation.

Experimental Setup: We compare EPP against MLM baselines using matched parameters. For 1000 WikiText-2 sentences, we measure cosine similarity between original and perturbed texts. For our EPP method we used the following parameters: $p \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$ and $m \in \{1, 5, 10, 15, 20, 25, 30, 35, 40, 50, 70\}$, and $k = 1$. For the MLM approach we used the model ‘distilbert-base-uncased’ with the same p values, $k = 1$. Since MLM-based perturbation is substantially more computationally expensive and does not admit a direct analog of the m parameter, we evaluate it on a randomly sampled subset of 100 WikiText-2 sentences, varying only p .

Key Findings: Figures 14 and 15 present the systematic comparison of EPP’s semantic preservation characteristics.

- **Higher semantic preservation:** EPP maintains higher semantic similarity compared to MLM-based perturbations across all parameter configurations. The embedding-based approach shows a smooth, controlled decrease in similarity as perturbation probability increases, as well as when expanding the neighbor range m . In contrast, MLM exhibits lower overall similarity, indicating more substantial semantic drift. Moreover, MLM exhibits more erratic patterns with unpredictable similarity drops.
- **Flexible similarity control:** EPP enables targeting both higher and lower similarity ranges compared to MLM. For conservative perturbations (small p , small m), EPP achieves similarity > 0.95 , preserving nearly identical semantic content. For more aggressive exploration (large p , large m), EPP can decrease similarity to ~ 0.75 - 0.85 while maintaining semantic coherence, providing a wider operational range

than MLM which plateaus around 0.80-0.90 regardless of parameters.

- **Computational efficiency:** EPP is substantially faster than MLM-based methods. Generating 100 neighbors with EPP takes ~ 0.5 seconds, while MLM requires ~ 10 -30 seconds depending on batch size and model complexity. This speedup enables efficient exploration of large neighborhood sets during loss landscape analysis.

Qualitative Analysis: To illustrate the practical differences, consider these representative examples with $p = 0.2$ and $m = 5$ for EPP vs. $m = 1$ for MLM:

- *Original:* “I’m going with my dog to the park this weekend...”
 - *EPP:* “I’m going with my dog with the parkThis weekend...” (sim: 0.97)
 - *MLM:* “I’m going with my family to central park this weekend...” (sim: 0.72)
- *Original:* “The quick brown fox jumps over the lazy dog near the riverbank...”
 - *EPP:* “The quick brown fox jumps over the lazy dog near The river banks...” (sim: 0.91)
 - *MLM:* “The quick brown fox jumps over the l-zy water near the riverbank...” (sim: 0.87)

These examples demonstrate that EPP produces minimal semantic drift by selecting embedding-proximate tokens (e.g., “riverbank” \rightarrow “river banks”), whereas MLM may introduce contextually plausible but semantically divergent replacements (e.g., “dog” \rightarrow “family”). A two-sample t-test confirms statistically significant differences ($p < 0.001$, Cohen’s $d > 0.8$), supporting EPP’s $30\times$ - $60\times$ speedup and improved semantic control for systematic loss landscape exploration.

B.3 Tokenizer Robustness

We evaluate the robustness of our memorization detection pipeline across three tokenizer sources: bert_cased, t5_based, and gpt2. The comparison keeps the core pipeline fixed ($top_k = 20$, $k_neighbors = 15$, $n_tokens = 0.6$, $MNT = 300$, $test_size = 0.2$, $reph = False$), so that

observed differences reflect tokenizer-induced variation rather than changes in the geometric feature extraction.

Key Findings: Figures 41, 42, 43, 44, and 45 present the systematic comparison of the tokenizers.

- **Consistent overall performance:** All three tokenizer sources yield highly consistent classification performance across metrics, indicating that our ILL-based features are stable under different tokenization schemes (Figures 41 and 42).
- **Stable Forget Set Distinguishability:** Identifying unlearned (forget) examples remains strongly robust across tokenizers. Aggregate mean AUCs for Forget vs. All are remarkably stable across the three sources: 0.829 for bert_cased, 0.834 for gpt2, and 0.812 for t5_based. Similarly, Forget vs. Hold-out mean AUCs are tightly clustered (0.877–0.888). Retain set metrics, such as Retain vs. All, likewise differ only modestly (≈ 0.83 –0.86).
- **Unlearning method invariant robustness:** Unlearning method level comparisons show that for strong unlearning baselines (e.g., ELM, RMU), the ability to detect the forget set is largely invariant to the specific tokenizer used. More challenging baselines like SimNPO exhibit slightly more tokenizer-dependent fluctuation but maintain relative stability across configurations.
- **Fine-grained stability:** Heatmaps breaking down performance by model and method natively (e.g., Figures 45, 46, and 47) reveal that while specific model-tokenizer pairs slightly influence absolute AUC for forget set detection, the rank-order of distinguishability remains strongly intact.

Implications: The tokenizer robustness analysis demonstrates that the geometric loss landscape signatures underlying our approach are not tightly coupled to a single tokenizer architecture. This supports the use of our method across diverse model families and tokenization schemes and suggests that the discriminative signals are driven by the overarching ILL geometry rather than distinct tokenizer artifacts.

B.4 Transferability Analysis

Experimental Setup: We evaluate whether classifiers trained on one model-dataset-method combination generalize to others. We examine: (1) *cross-unlearning* (different unlearning method), (2) *cross-model* (different architecture), (3) *cross-dataset* (different dataset), and (4) *full cross-transfer* (both different).

Key Findings: Figures 11, 12, and 13 present evidence for cross-context generalization.

- **Cross-unlearning performance:** Classifiers trained on one unlearning method achieve high accuracy when tested on different unlearning methods applied to the same model-dataset pair, demonstrating that geometric signatures generalize across unlearning algorithms (Figure 39). Importantly, transferability provides insights into the nature of unlearning frameworks:
 - *Specialist vs. Generalist Geometry:* Certain unlearning techniques induce broader or more specialized input loss landscapes. For instance, testing a classifier trained on the `rmu-lat` unlearned model against an `rmu` model yields near-perfect transferability (96.1 AUC). This strong affinity underscores that `rmu-lat` acts as a specialized superset or variation of `rmu`, fully retaining its geometric properties. Conversely, `rmu-lat` generalizes poorly to distinct unlearning methods like `tar` (46.3) or `pbj` (47.6), indicating a high degree of specialization. In contrast, training on `tar` and transferring back to `pbj` shows relatively broad generalist transferability (82.8 AUC).
 - *Asymmetric Transferability:* The transfer of geometric features is frequently not symmetric. Training on `tar` and testing on `pbj` results in a strong score of 82.8, while reversing the arrangement (training on `pbj` and testing on `tar`) drops performance heavily to 59.1. This asymmetry implies that some unlearning methods (e.g., `tar`) entail geometric modifications that strictly encompass those produced by other methods.
- **Cross-model transfer:** When transferring classifiers across different model architectures (e.g., from LLaMA3 to zephyr or to

LLaMA2), performance retention varies substantially, with accuracy average of 31-67% depending on model similarity. Models with similar architectures (e.g., LLaMA variants) transfer better than more distinct ones (e.g., zephyr). Note that same model transfers achieve higher but not perfect scores, because of averaging across different datasets and unlearning methods.

- **Moderate cross-dataset transfer:** Transfer across datasets shows greater variation, with performance retention ranging from 60-85% depending on dataset similarity. Figure 12 shows that semantically similar datasets (e.g., both text generation tasks) transfer better than semantically distant ones.
- **Performance distribution analysis:** Figure 11 shows that the majority of transfer scenarios maintain strong performance, with a clear separation between successful transfers (AUC > 0.8) and challenging cross-domain transfers.
- **Consistent neighborhood parameter effects:** The transferability patterns hold across different values of `n_tokens`, `max_neighbors`, and `neighbor_dist`, indicating that the underlying geometric signatures are robust to hyperparameter choices during both training and transfer.

These results demonstrate that geometric signatures reflect fundamental memorization properties that transcend specific architectures and datasets, enabling practical one-time training with cross-context deployment.

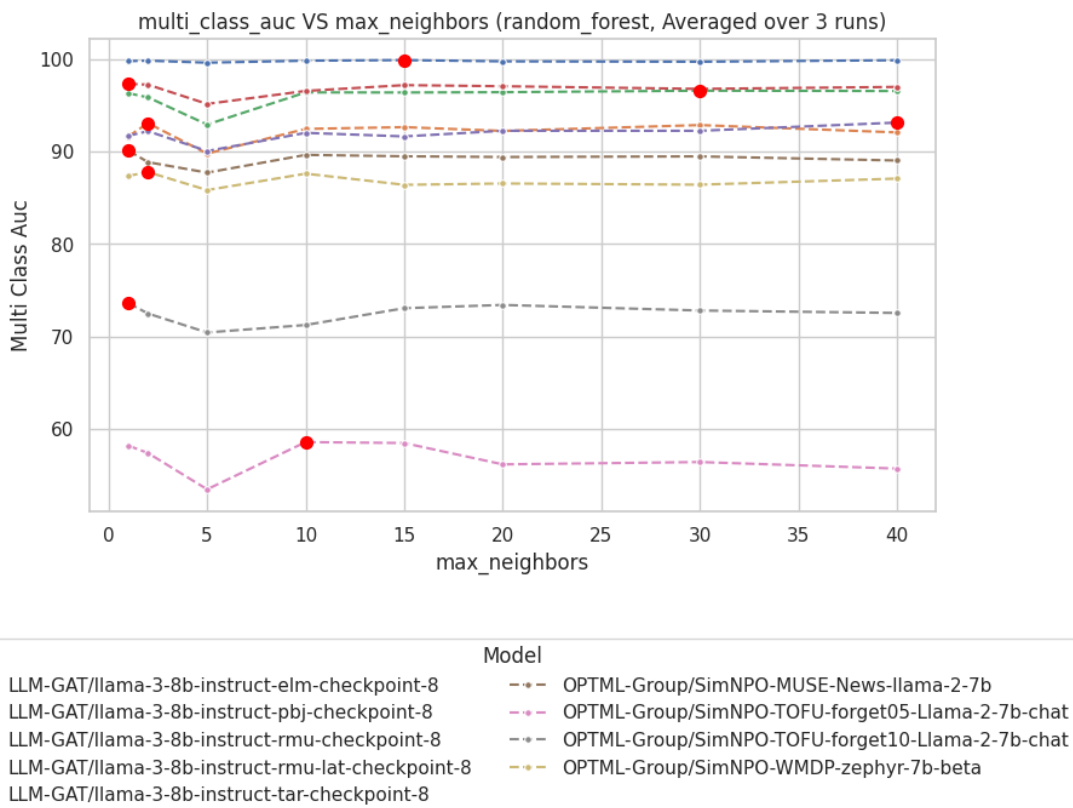


Figure 8: **Sensitivity to number of sampled variants (K).** Multi-class classification performance (ROC-AUC) remains stable across a wide range of K values, demonstrating that REMIND captures meaningful geometric signatures even with small neighborhoods. Red dots indicate maximum performance values.

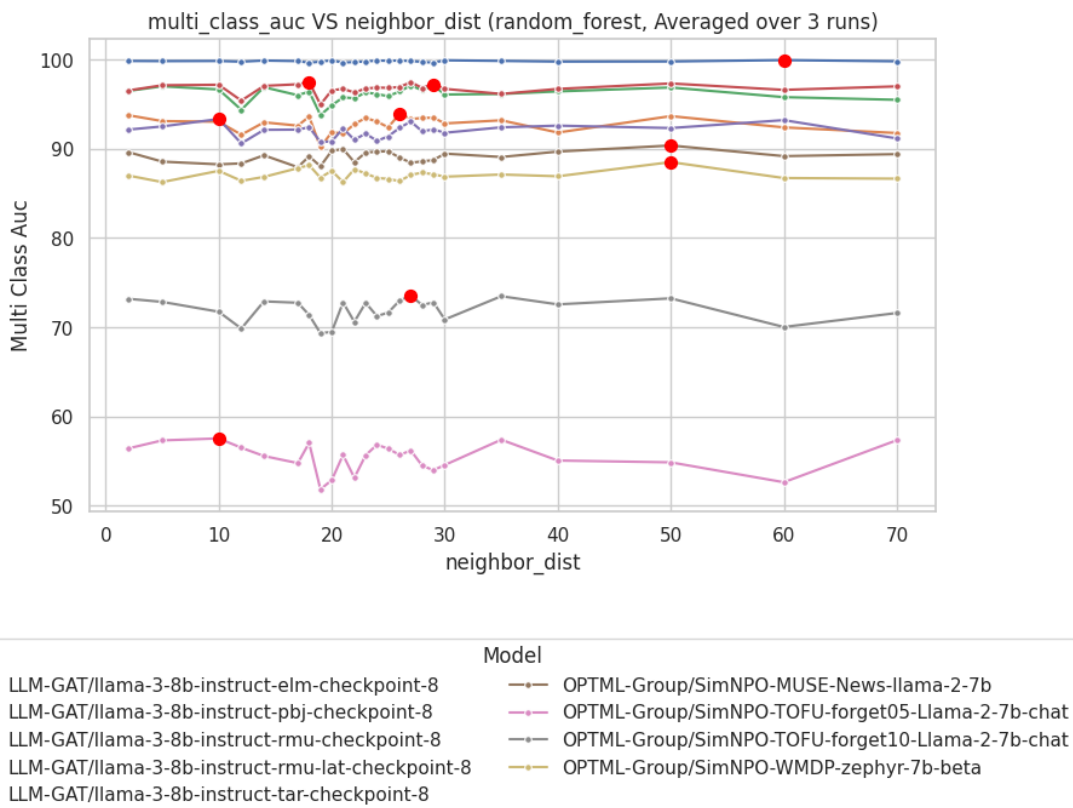


Figure 9: **Sensitivity to embedding-space neighbor range (m)**. Classification performance remains robust across varying semantic radii, with AUC scores stable from tight paraphrases ($m = 2$) to broader semantic neighborhoods ($m = 70$). Red dots indicate maximum performance values.

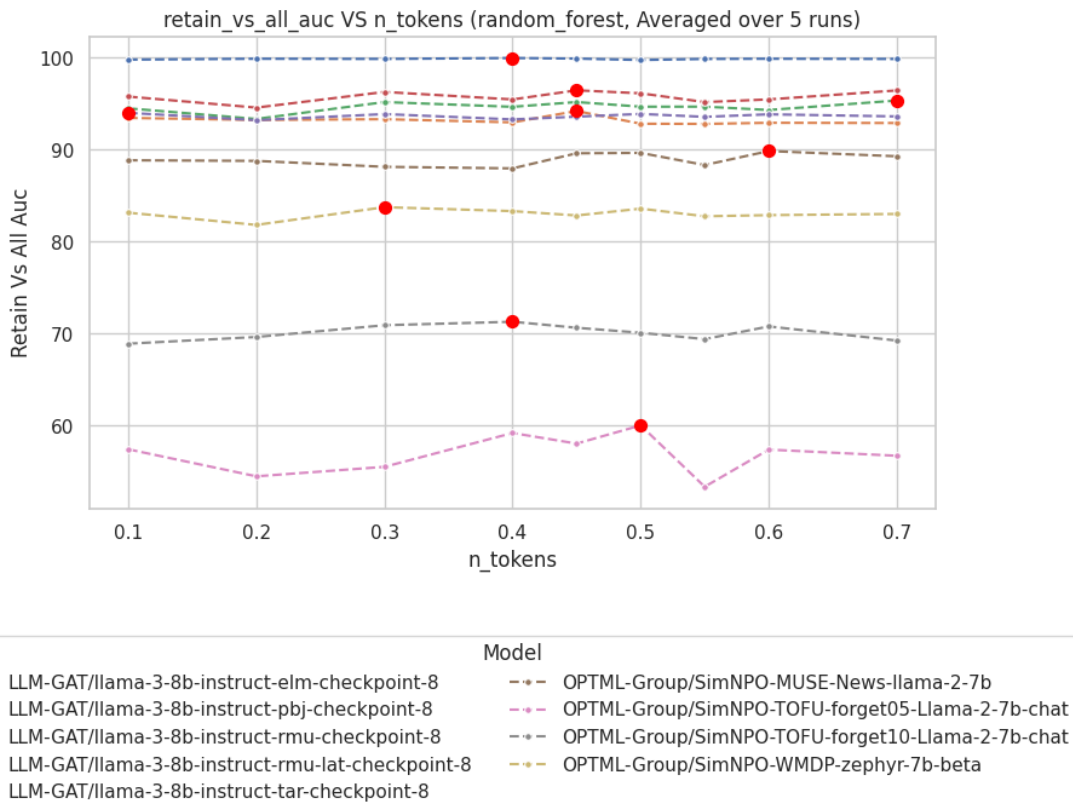


Figure 10: **Sensitivity to perturbation probability (p)**. Retain vs. all classification maintains high AUC across diverse perturbation levels, indicating that geometric signatures persist regardless of perturbation magnitude. Red dots indicate maximum performance values.

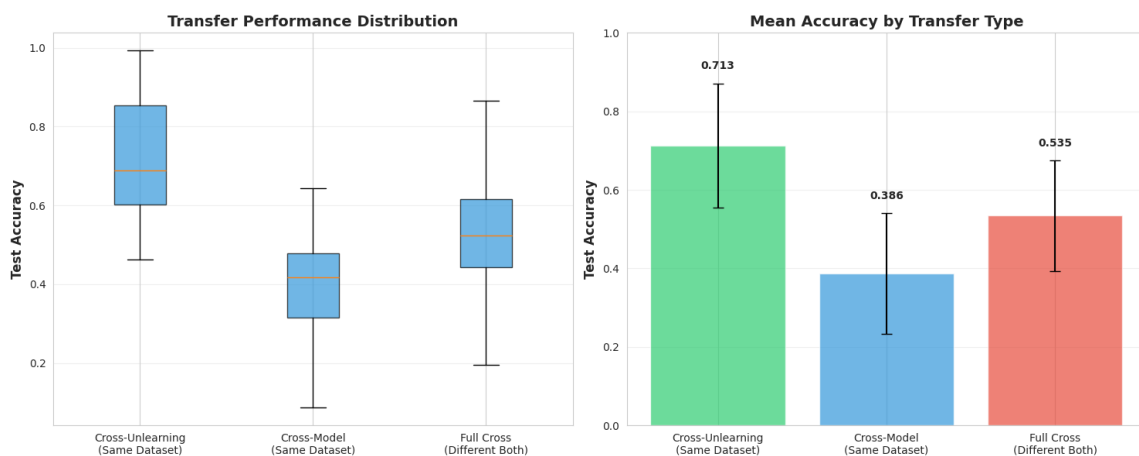


Figure 11: **Transfer performance distribution**. Distribution of test accuracy when transferring classifiers across different model-dataset combinations, showing strong generalization with most transfers maintaining high performance.

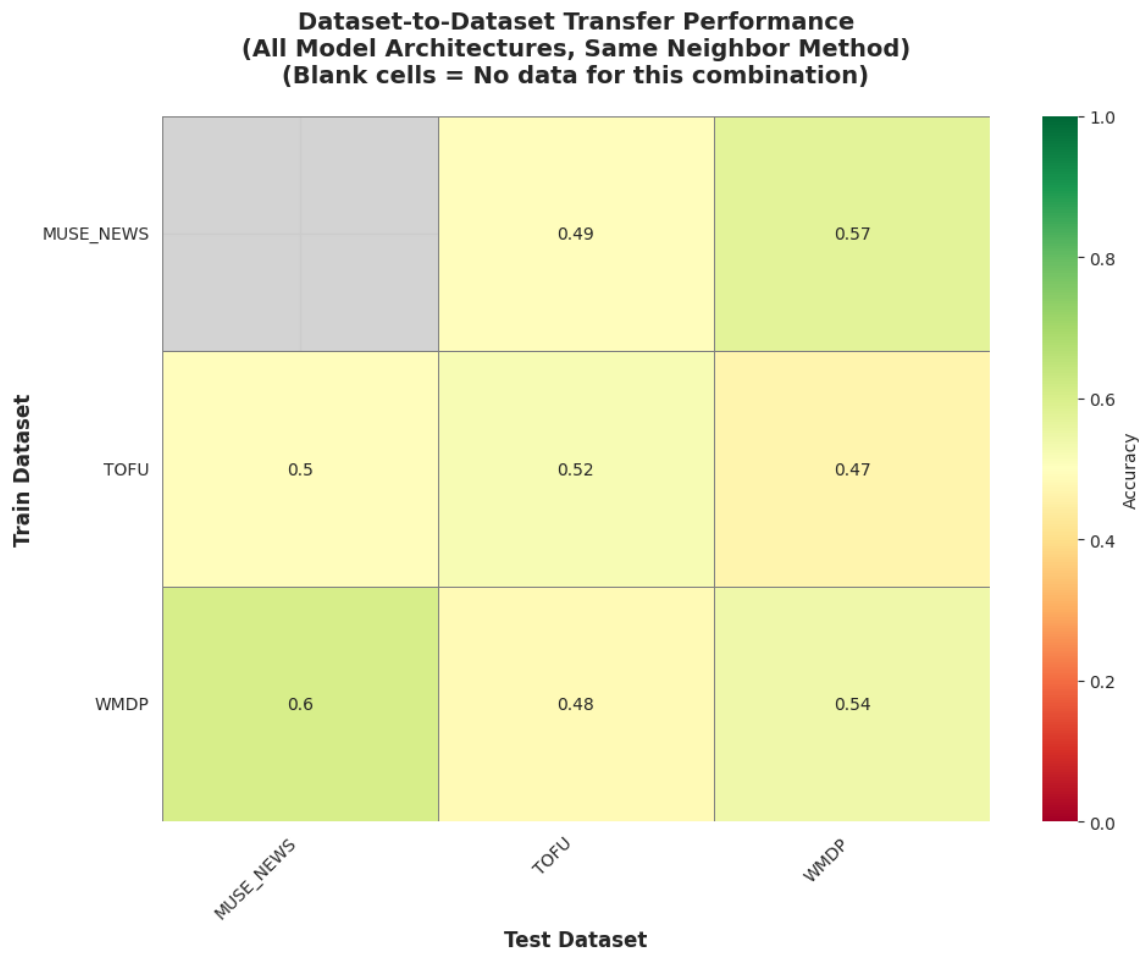


Figure 12: **Cross-dataset transfer heatmap.** Performance when transferring classifiers trained on one dataset to another, revealing that semantic similarity between datasets predicts transfer success. Grey cells indicate transfer scenarios that were not tested due to data unavailability.

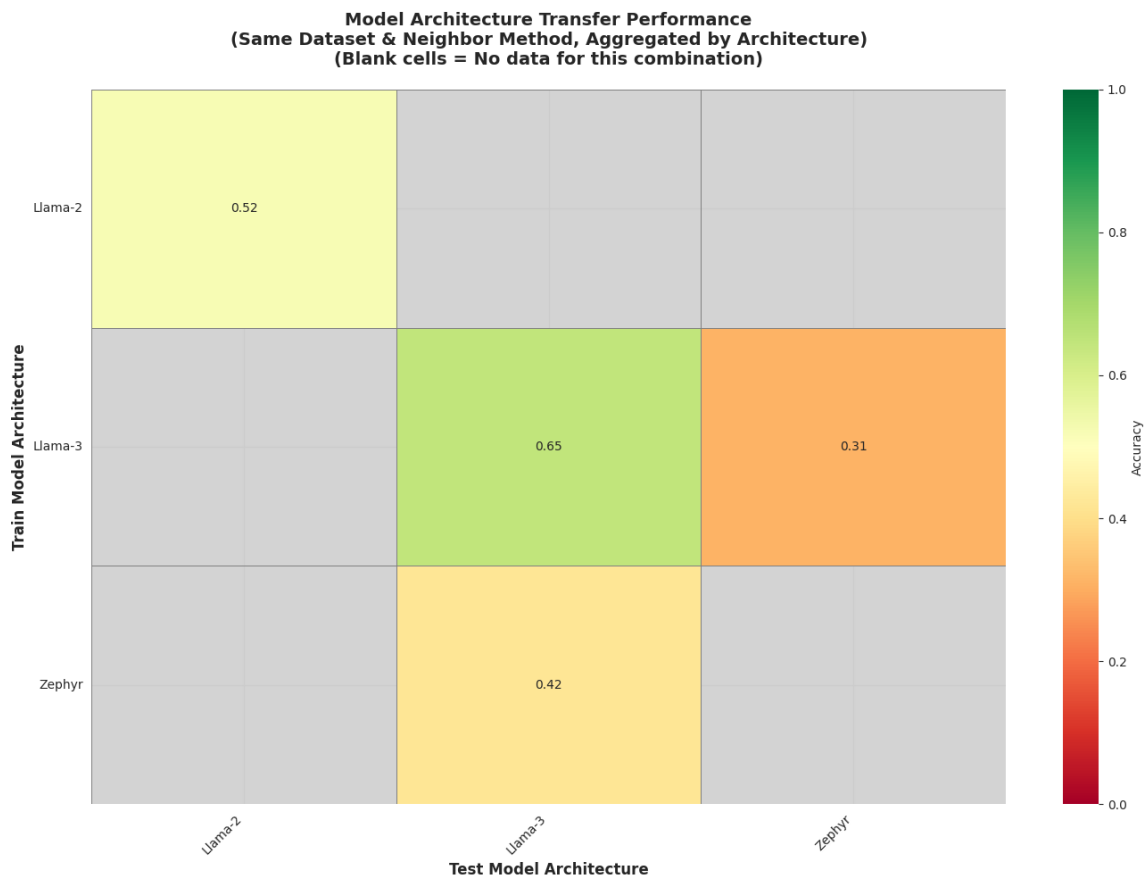


Figure 13: **Cross-model architecture transfer heatmap.** AUC scores when transferring classifiers between model architectures, demonstrating robust cross-architecture generalization. Grey cells indicate transfer scenarios that were not tested due to data unavailability.

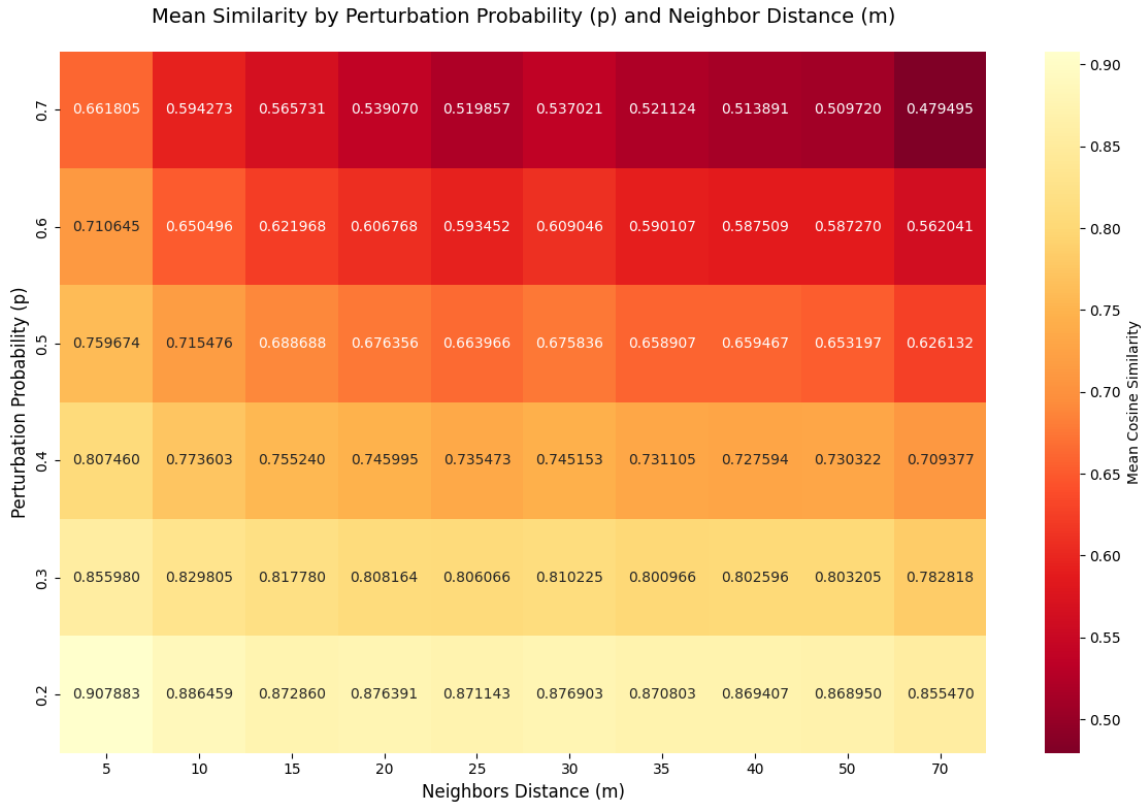


Figure 14: **EPP semantic preservation across parameter space.** Heatmap showing mean cosine similarity between original and EPP-perturbed texts as a function of perturbation probability p and neighbor distance m . The smooth gradient demonstrates predictable, controllable semantic variation, with similarity decreasing monotonically as perturbation strength increases.

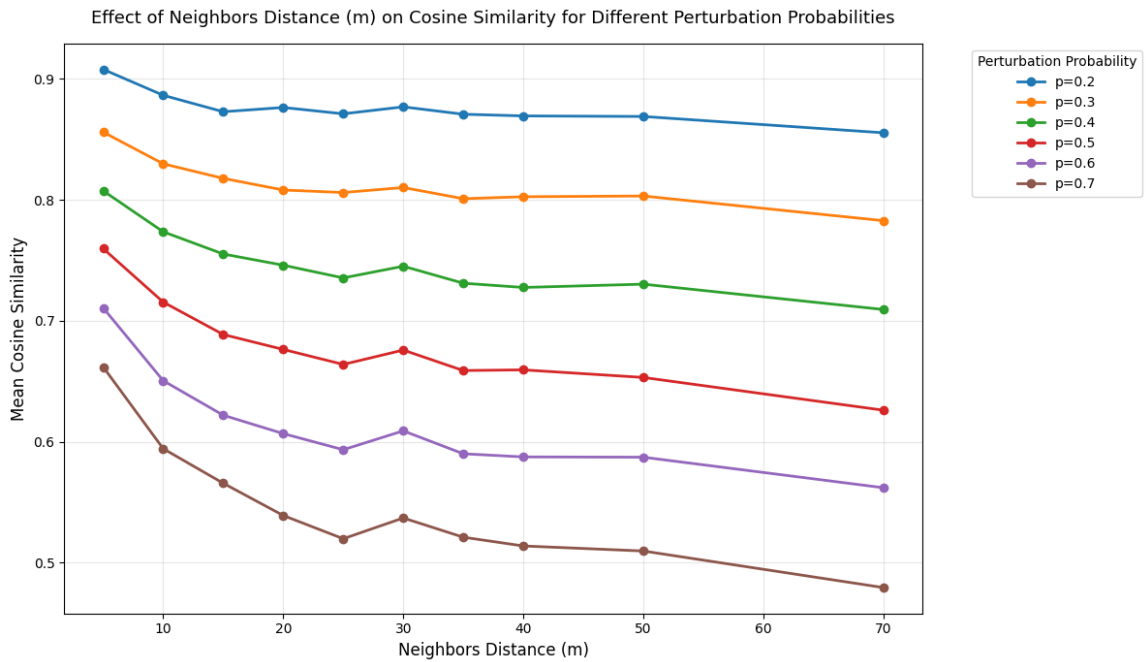


Figure 15: **Effect of neighbor distance on semantic similarity.** Line plots showing how mean cosine similarity varies with embedding-space neighbor range m for different perturbation probabilities p . EPP exhibits smooth, monotonic behavior, enabling predictable control over the semantic exploration radius.

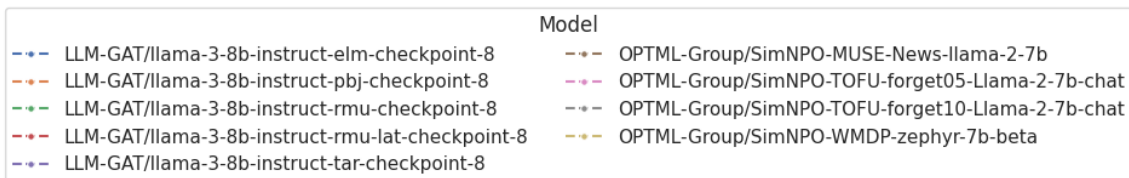
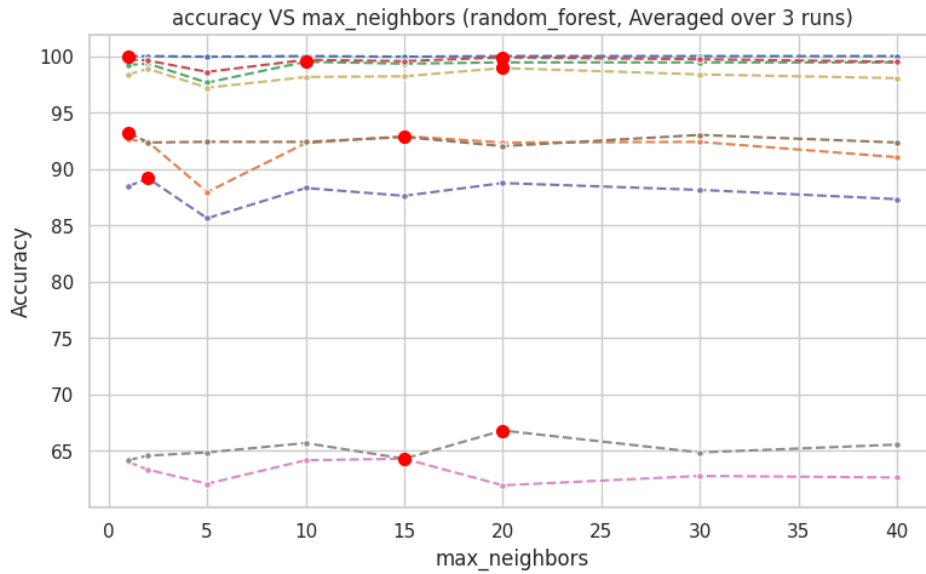


Figure 16: **Classification accuracy vs. number of sampled variants (K).** Overall classification accuracy across memorization states as a function of neighborhood sample size.

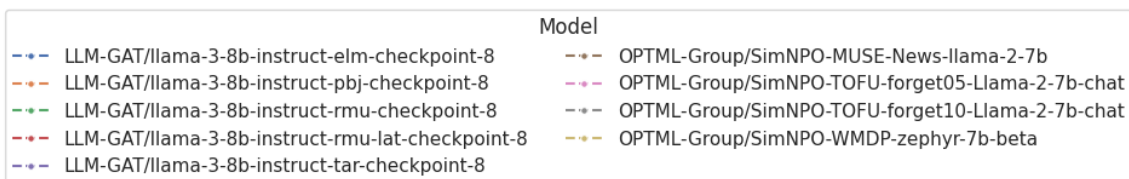
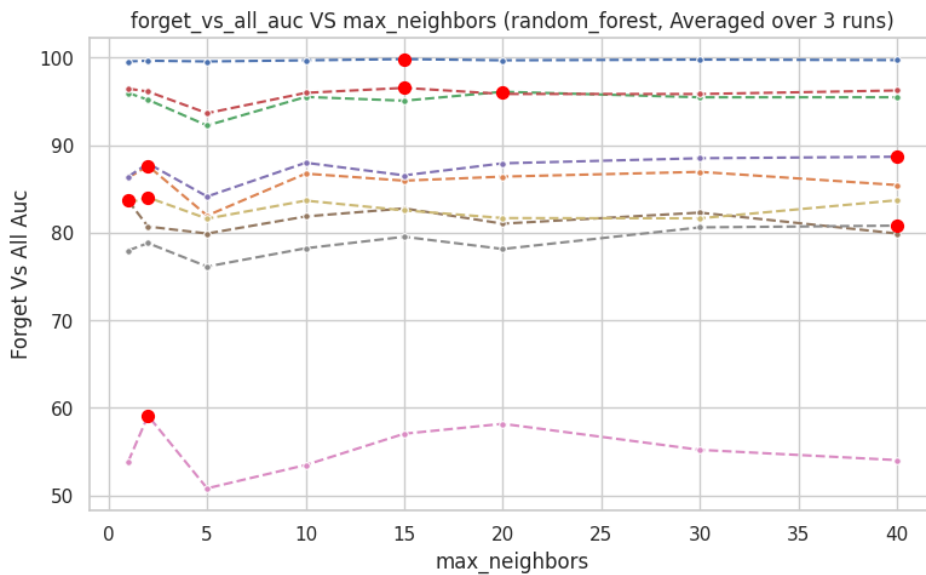


Figure 17: **Forget vs. all classification with varying K .** AUC performance for distinguishing forget examples from all other examples across different neighborhood sizes.

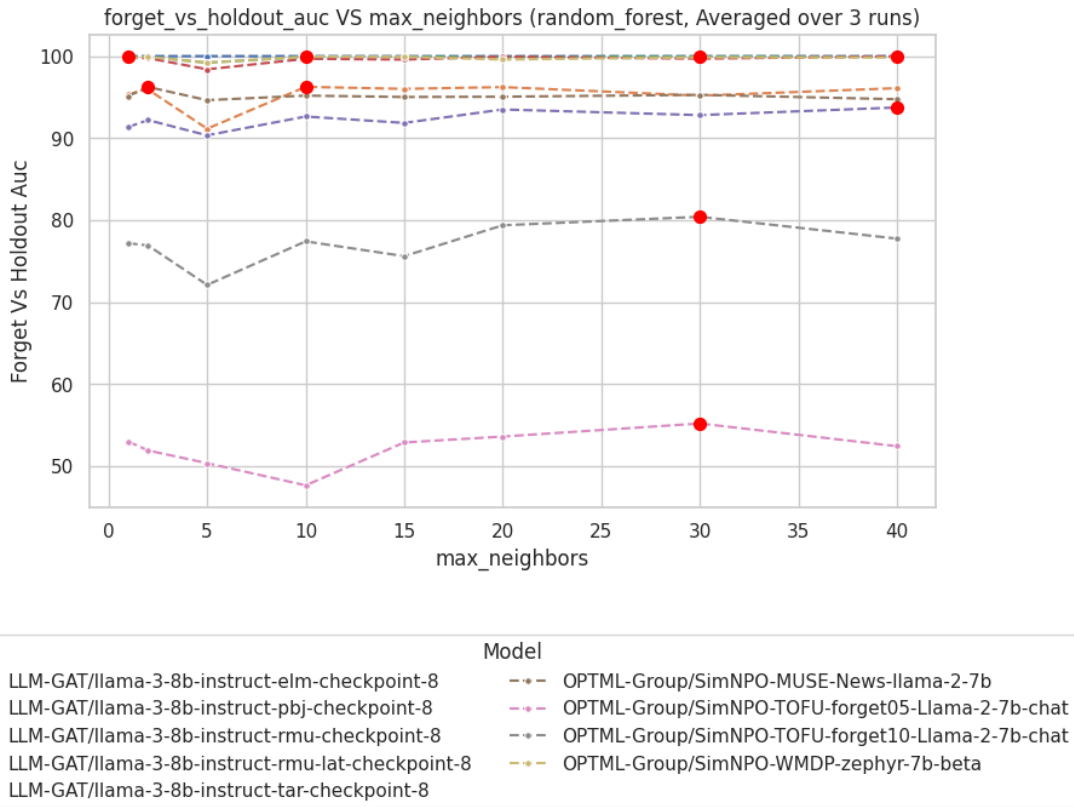


Figure 18: **Forget vs. holdout classification with varying K .** AUC performance for discriminating forget from holdout examples across different neighborhood sizes.

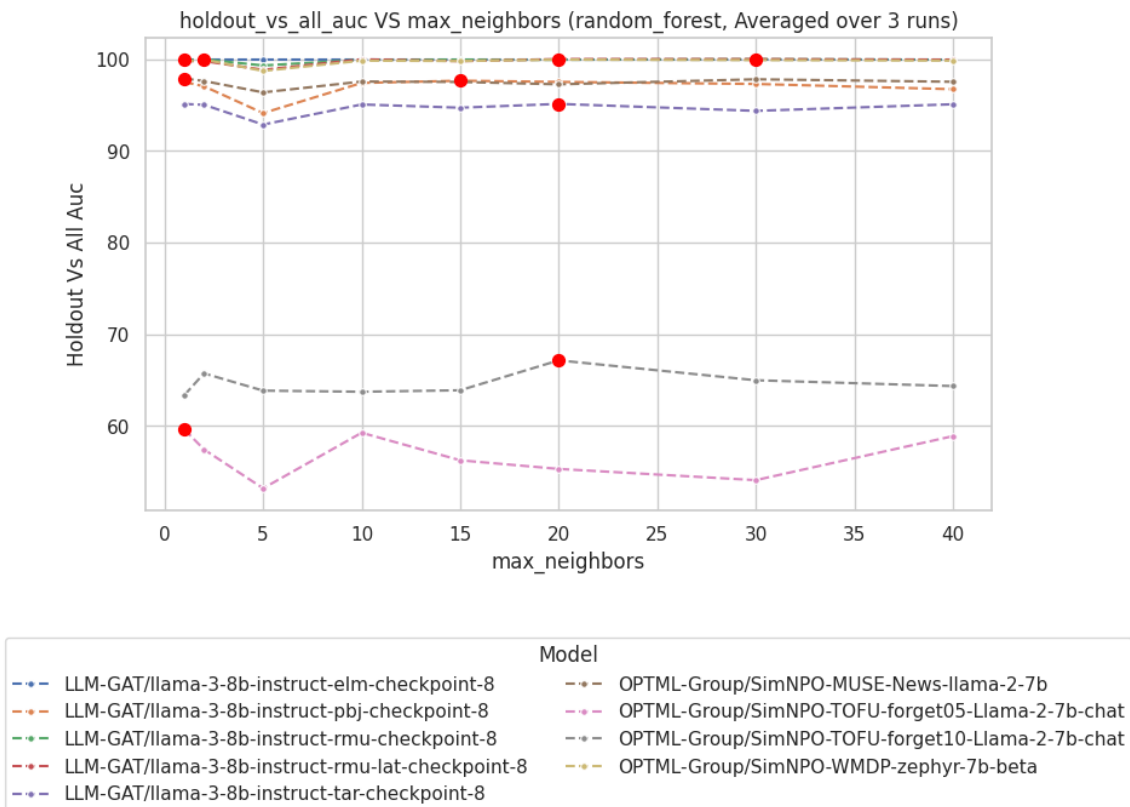


Figure 19: **Holdout vs. all classification with varying K .** AUC performance for distinguishing holdout examples from all other examples across different neighborhood sizes.

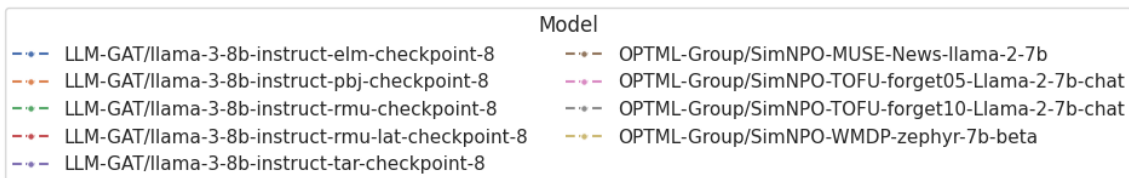
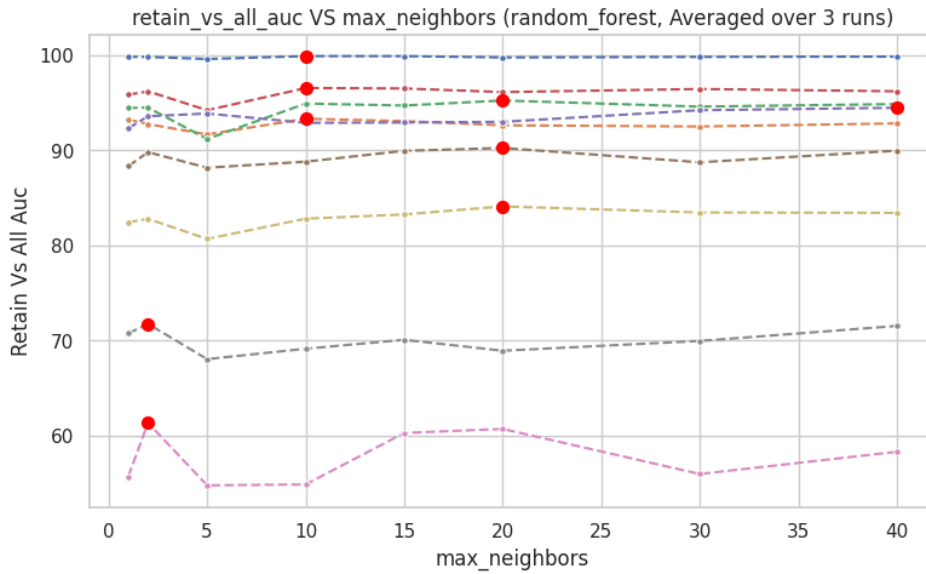


Figure 20: **Retain vs. all classification with varying K .** AUC performance for distinguishing retain examples from all other examples across different neighborhood sizes.

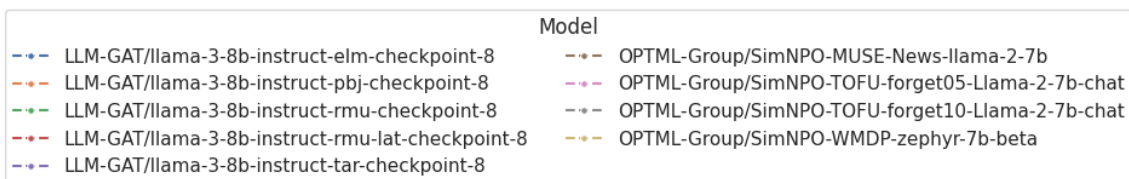
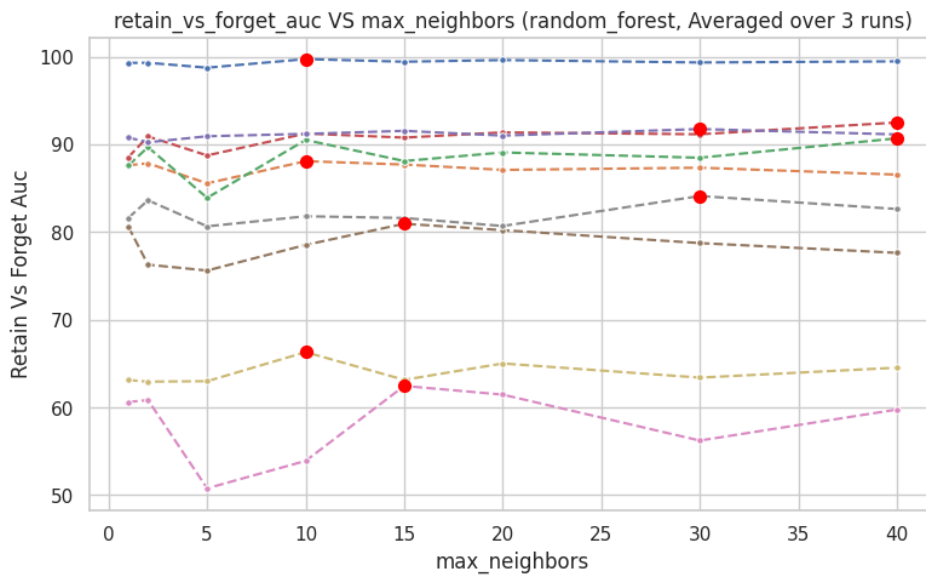


Figure 21: **Retain vs. forget classification with varying K .** AUC performance for discriminating retain from forget examples across different neighborhood sizes.

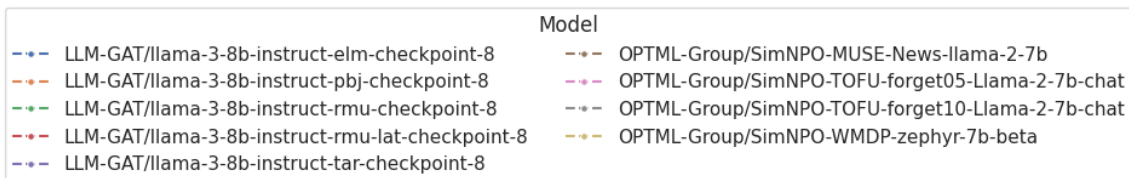
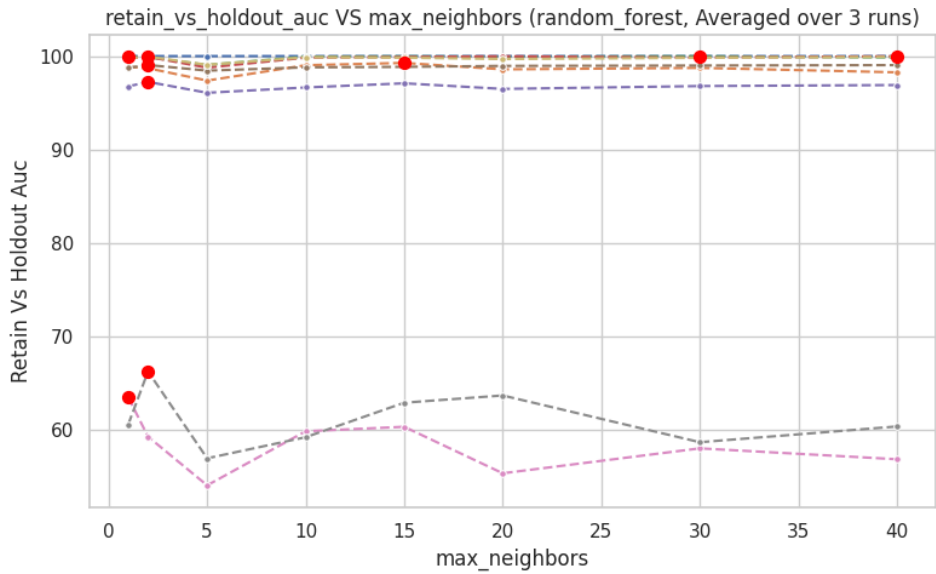


Figure 22: **Retain vs. holdout classification with varying K .** AUC performance for discriminating retain from holdout examples across different neighborhood sizes.

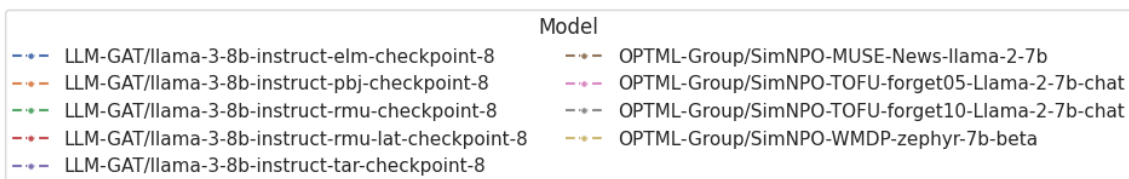
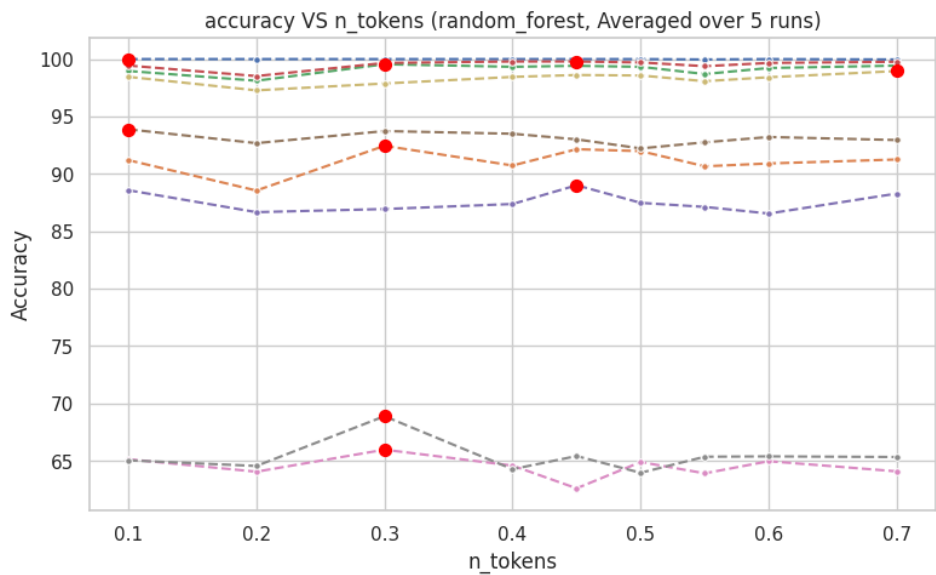


Figure 23: **Classification accuracy vs. perturbation probability (p).** Overall classification accuracy across memorization states as a function of perturbation strength.

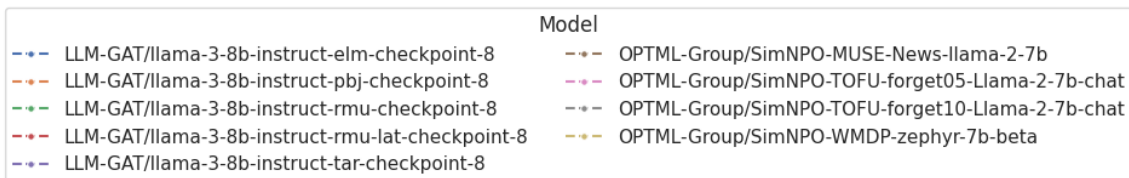
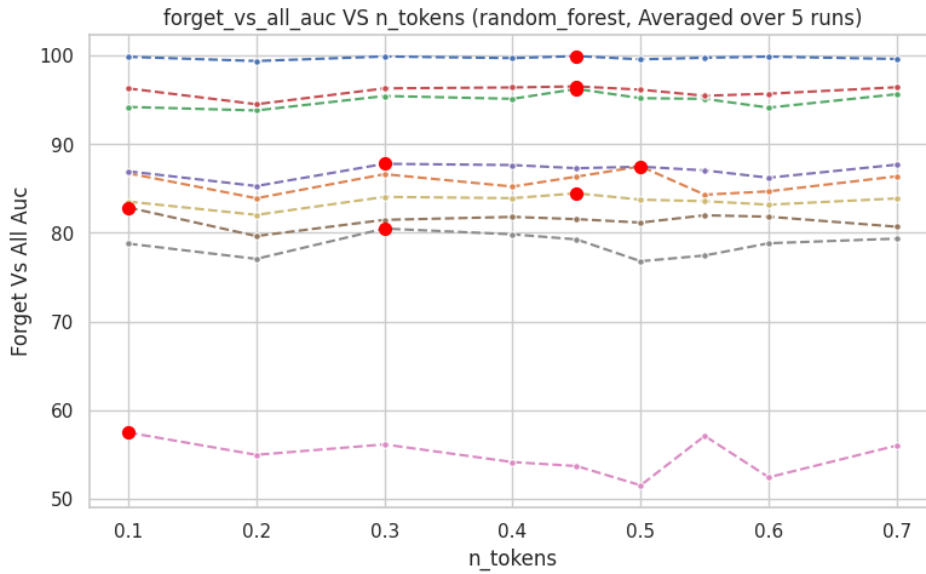


Figure 24: **Forget vs. all classification with varying p .** AUC performance for distinguishing forget examples from all other examples across different perturbation probabilities.

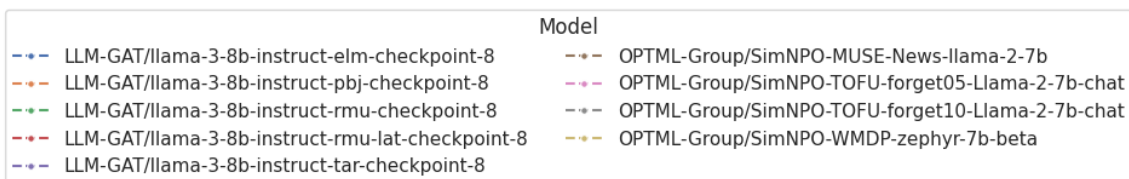
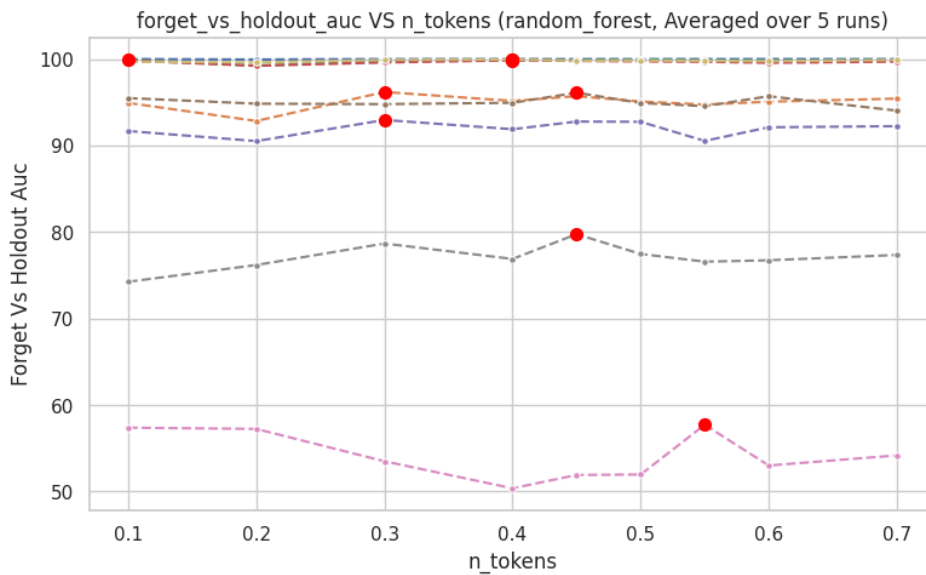


Figure 25: **Forget vs. holdout classification with varying p .** AUC performance for discriminating forget from holdout examples across different perturbation probabilities.

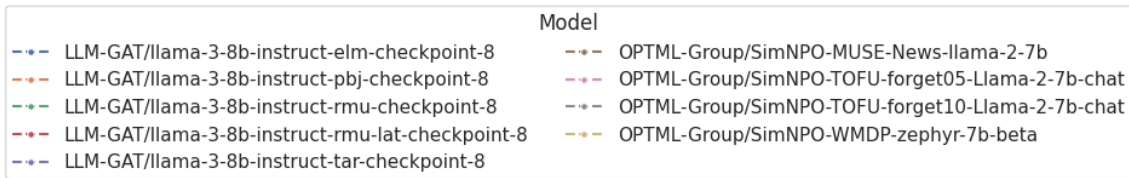
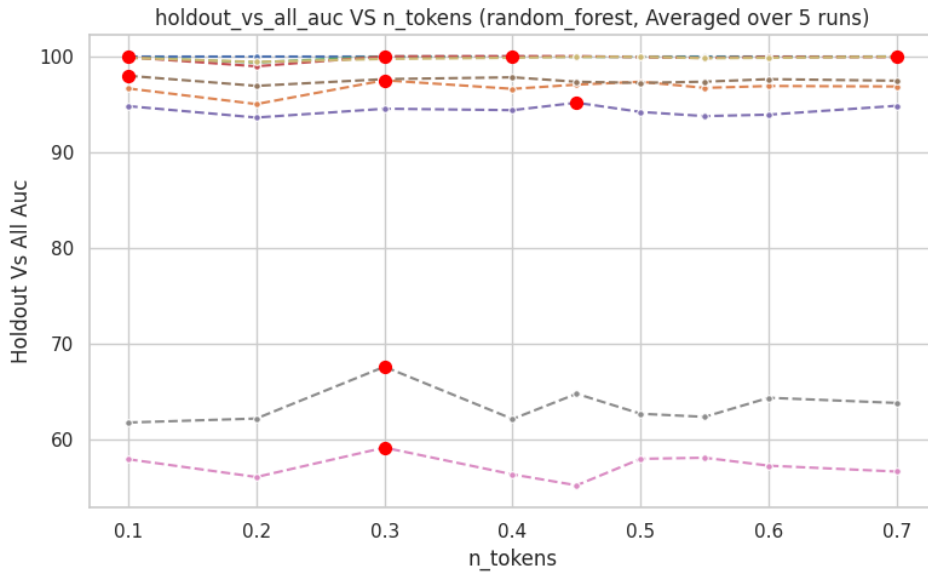


Figure 26: **Holdout vs. all classification with varying p .** AUC performance for distinguishing holdout examples from all other examples across different perturbation probabilities.

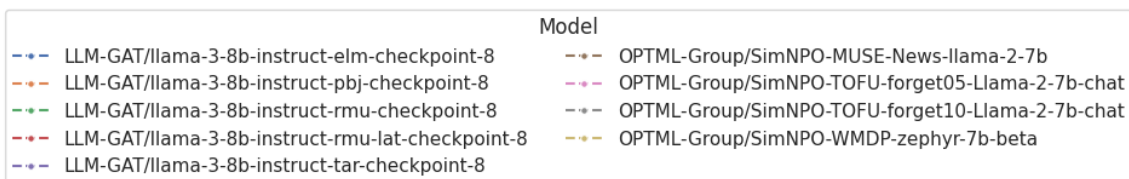
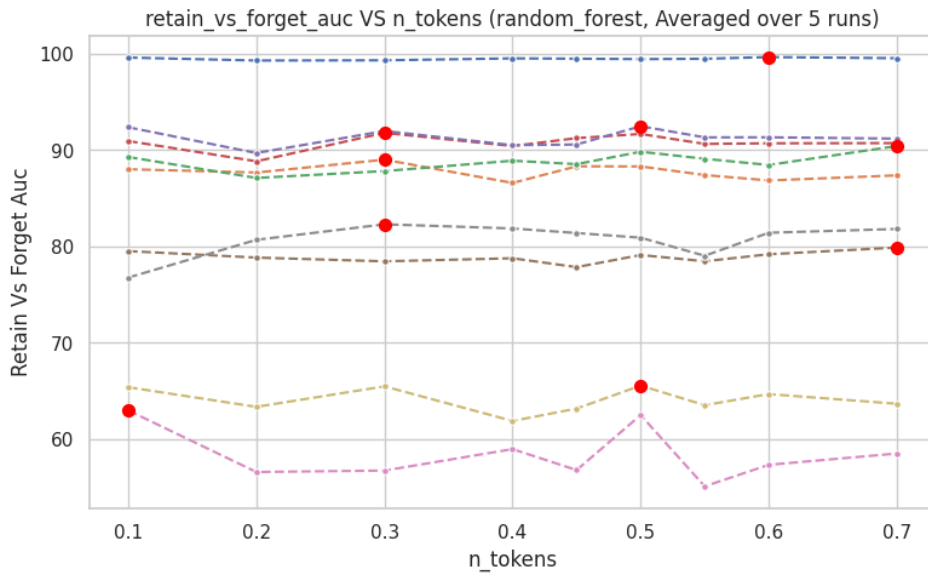


Figure 27: **Retain vs. forget classification with varying p .** AUC performance for discriminating retain from forget examples across different perturbation probabilities.

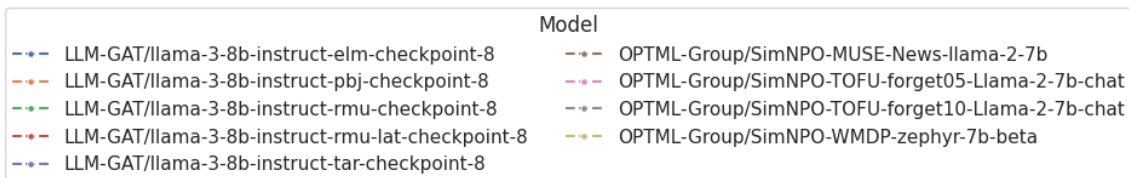
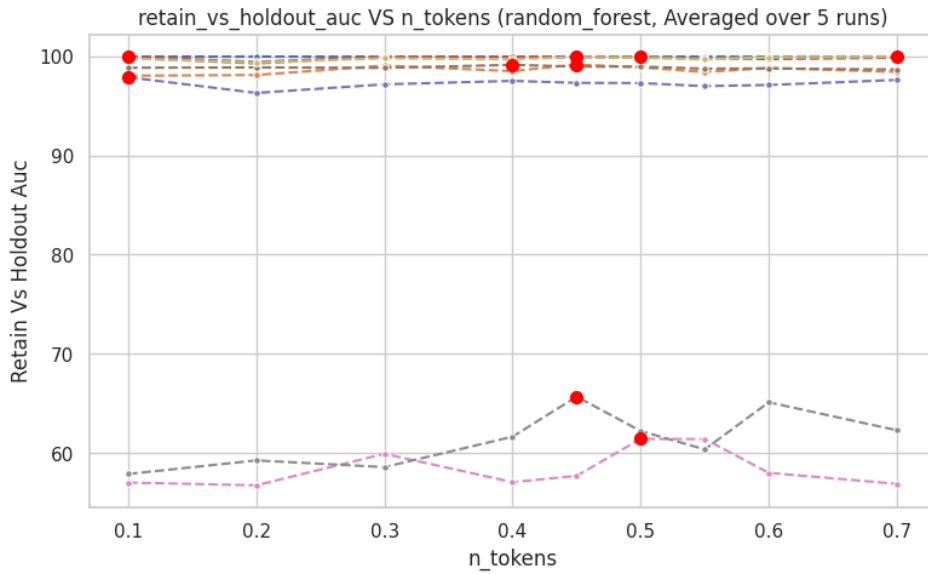


Figure 28: **Retain vs. holdout classification with varying p .** AUC performance for discriminating retain from holdout examples across different perturbation probabilities.

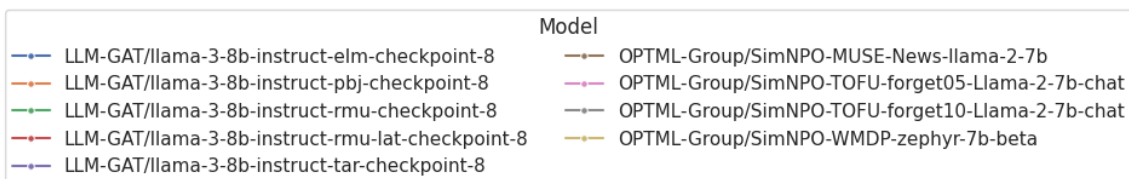
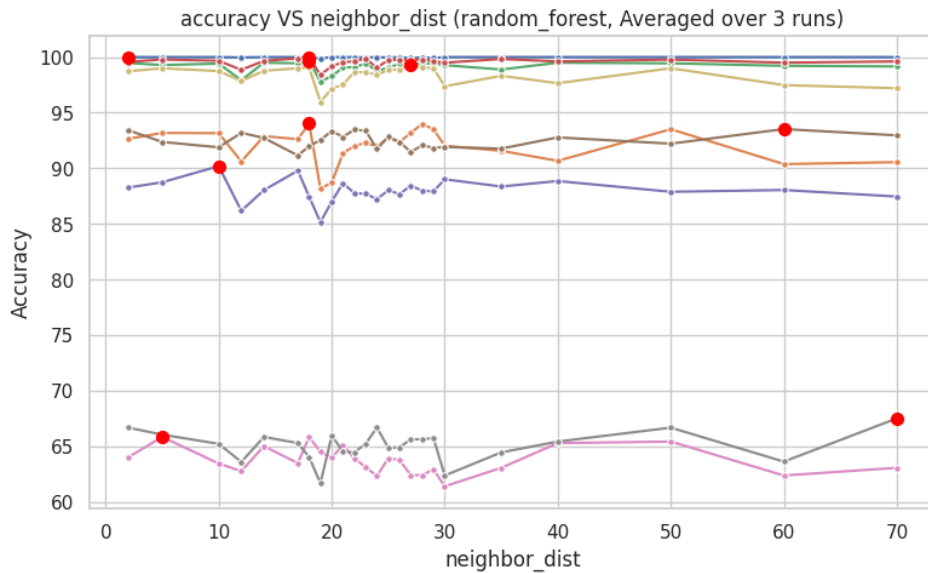


Figure 29: **Classification accuracy vs. embedding-space neighbor range (m).** Overall classification accuracy across memorization states as a function of semantic radius.

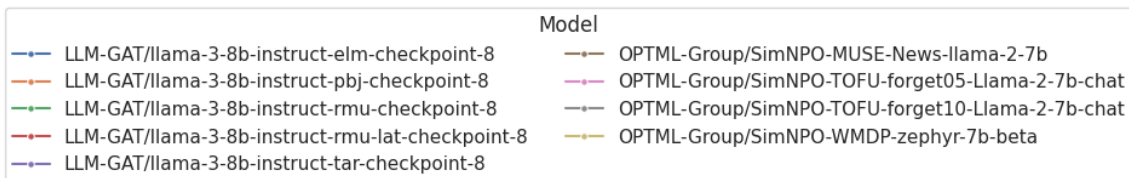
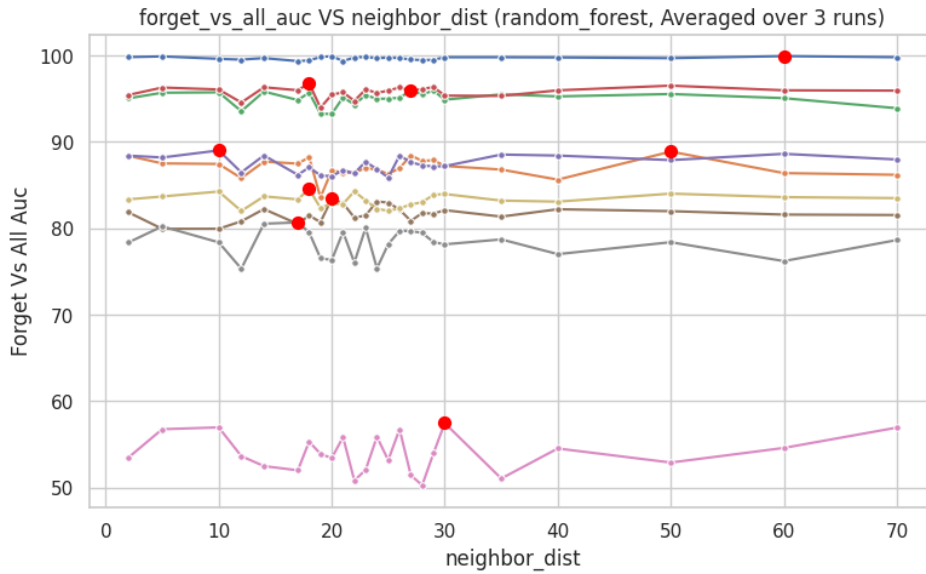


Figure 30: **Forget vs. all classification with varying m .** AUC performance for distinguishing forget examples from all other examples across different semantic radii.

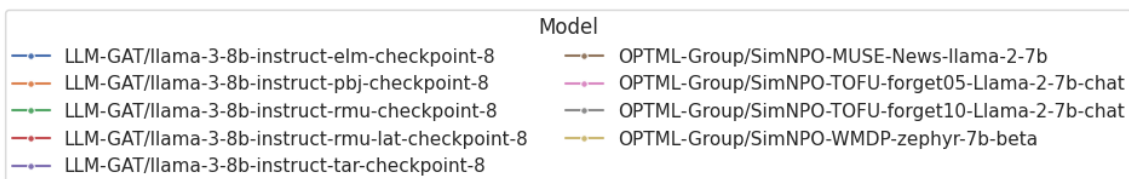
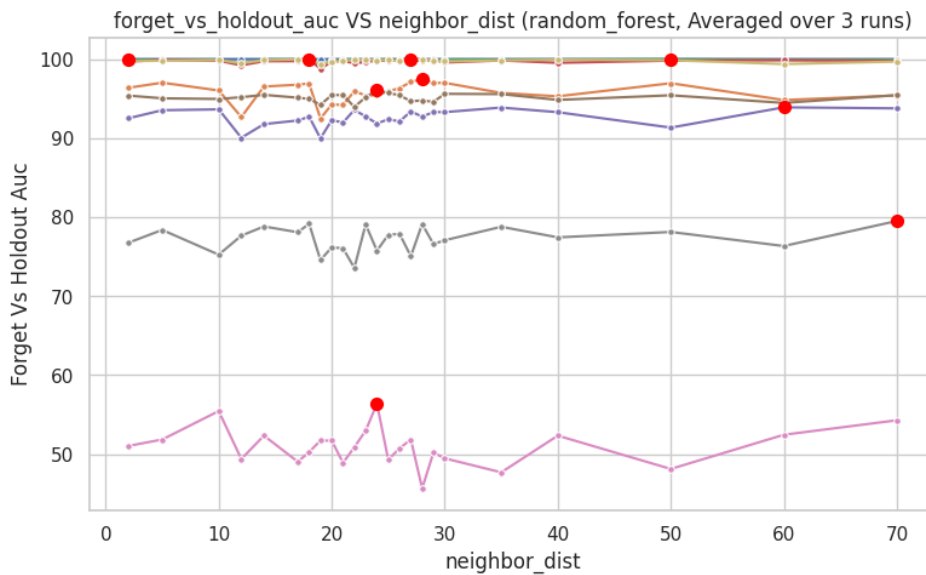


Figure 31: **Forget vs. holdout classification with varying m .** AUC performance for discriminating forget from holdout examples across different semantic radii.

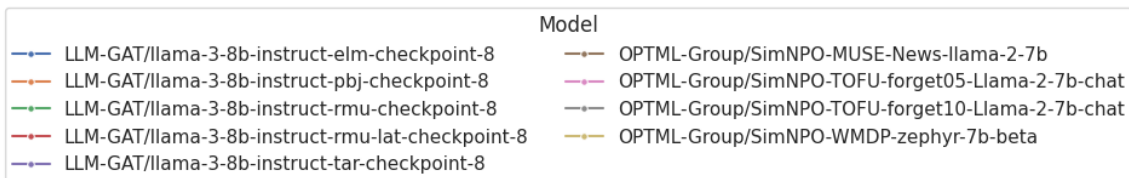
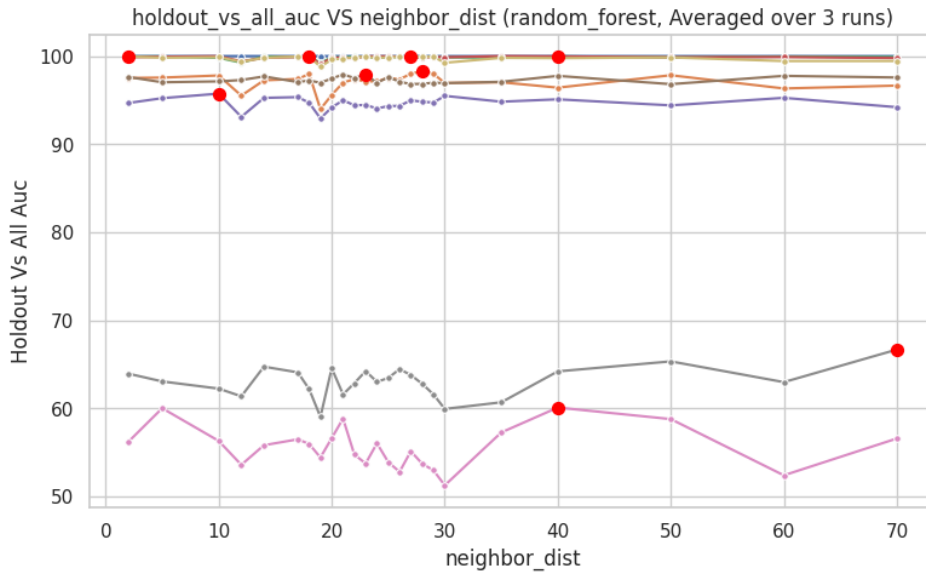


Figure 32: **Holdout vs. all classification with varying m .** AUC performance for distinguishing holdout examples from all other examples across different semantic radii.

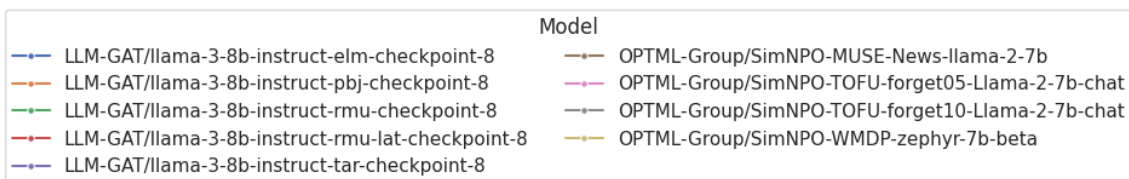
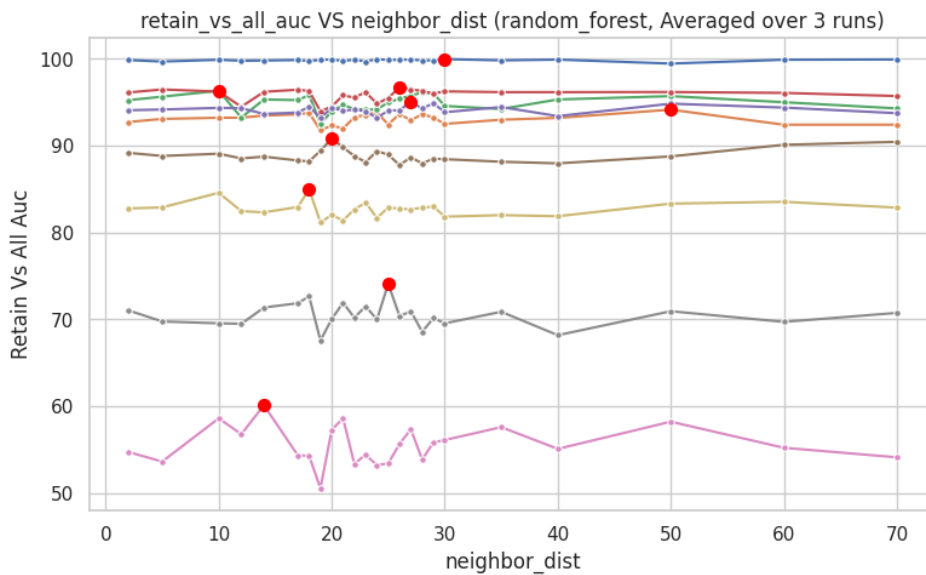


Figure 33: **Retain vs. all classification with varying m .** AUC performance for distinguishing retain examples from all other examples across different semantic radii.

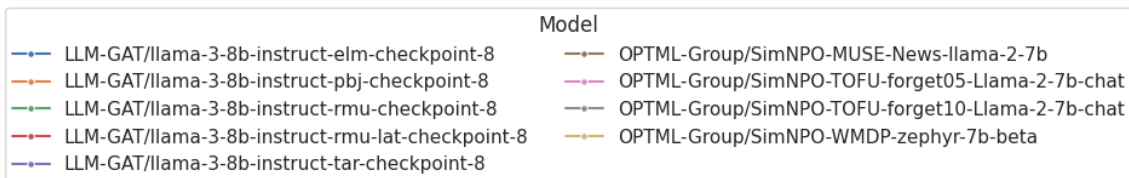
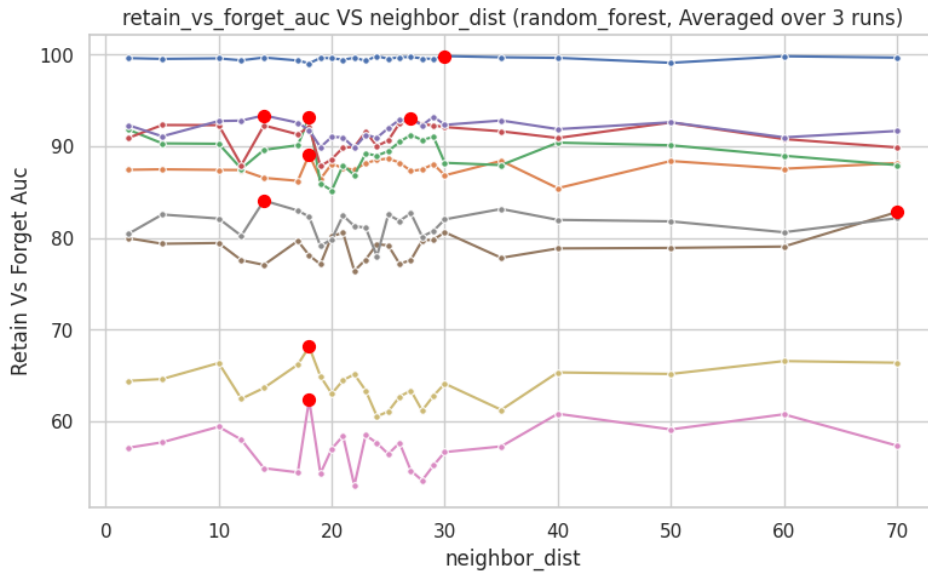


Figure 34: **Retain vs. forget classification with varying m .** AUC performance for discriminating retain from forget examples across different semantic radii.

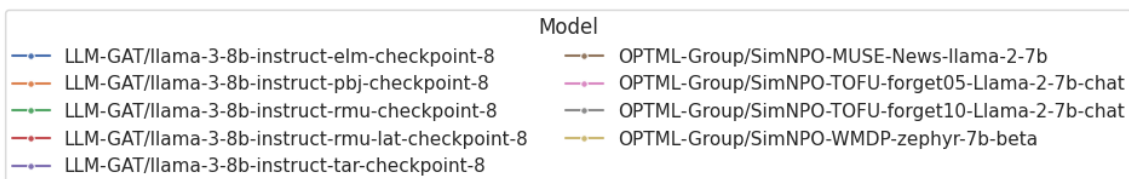
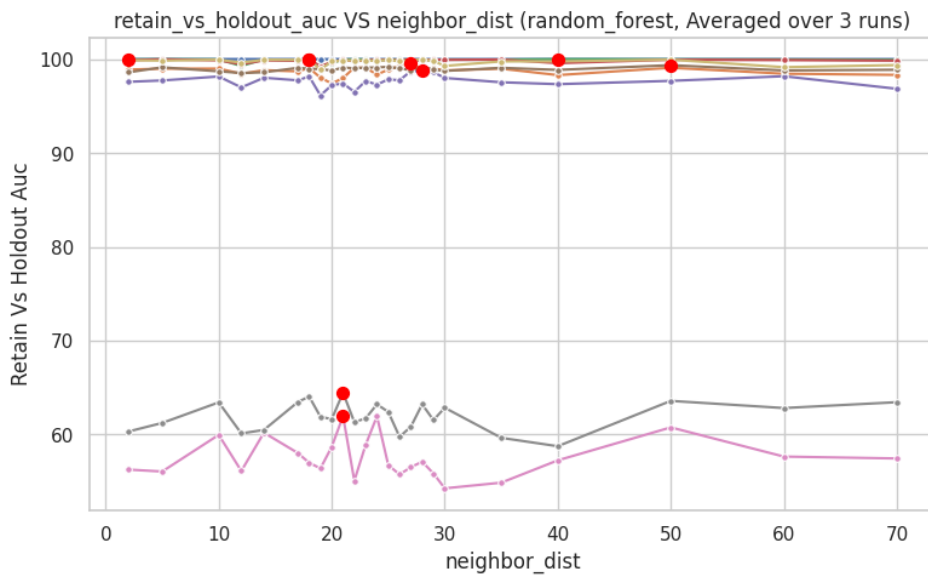


Figure 35: **Retain vs. holdout classification with varying m .** AUC performance for discriminating retain from holdout examples across different semantic radii.

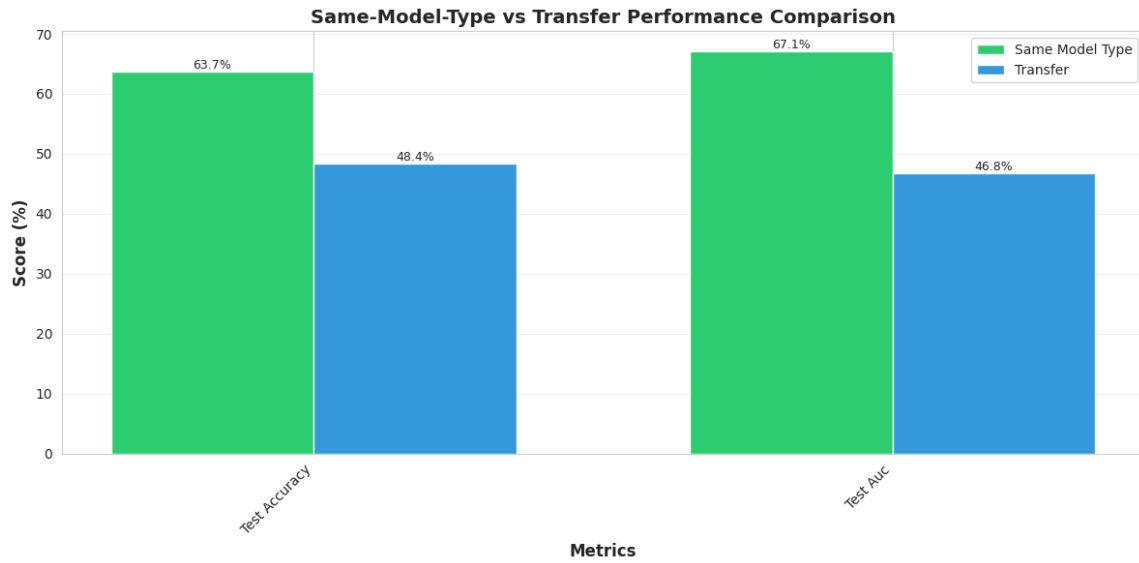


Figure 36: **Same model type vs. transfer performance.** Bar plot comparing classifier performance when trained and tested on the same model type versus cross-model transfer scenarios.

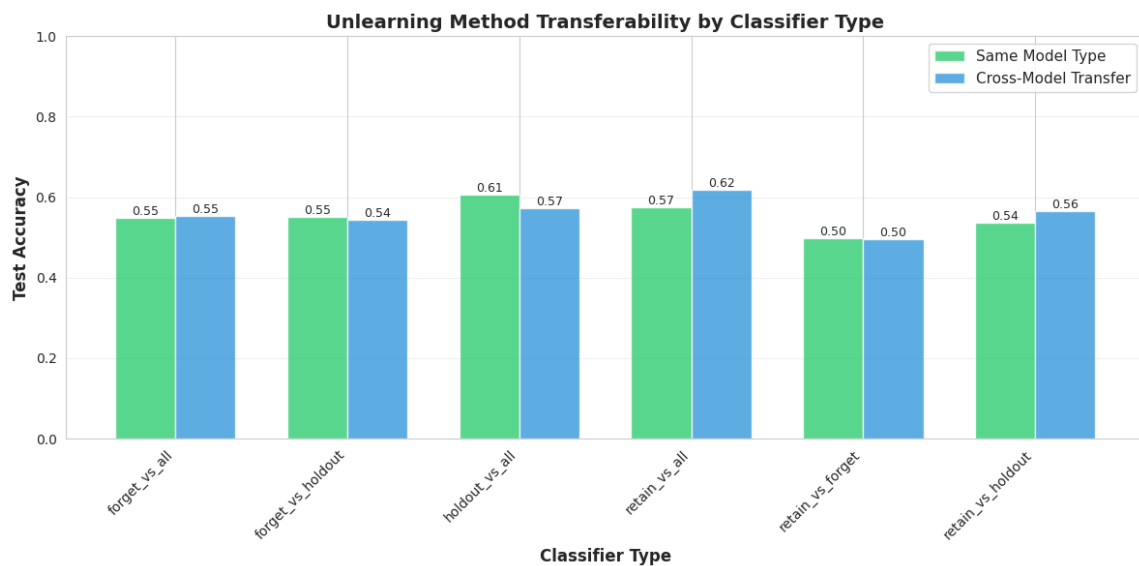


Figure 37: **Detailed transfer metrics comparison.** Bar plot showing comprehensive performance metrics across different transfer scenarios, breaking down classification accuracy by metric type.

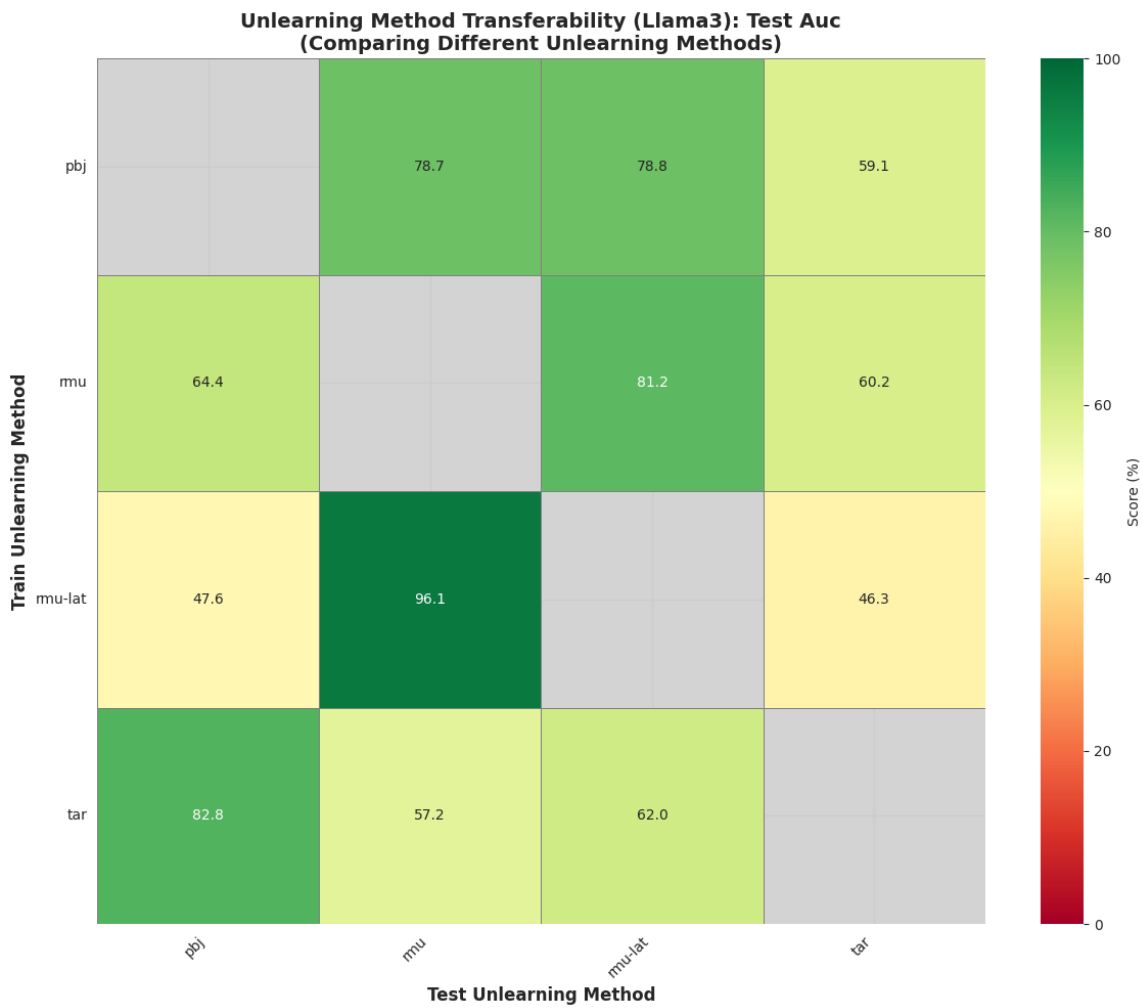


Figure 38: **Cross-unlearning method transfer heatmap.** Performance when transferring classifiers trained on one unlearning method to another, demonstrating generalization across unlearning algorithms. Grey cells indicate transfer scenarios that were not tested due to data unavailability.

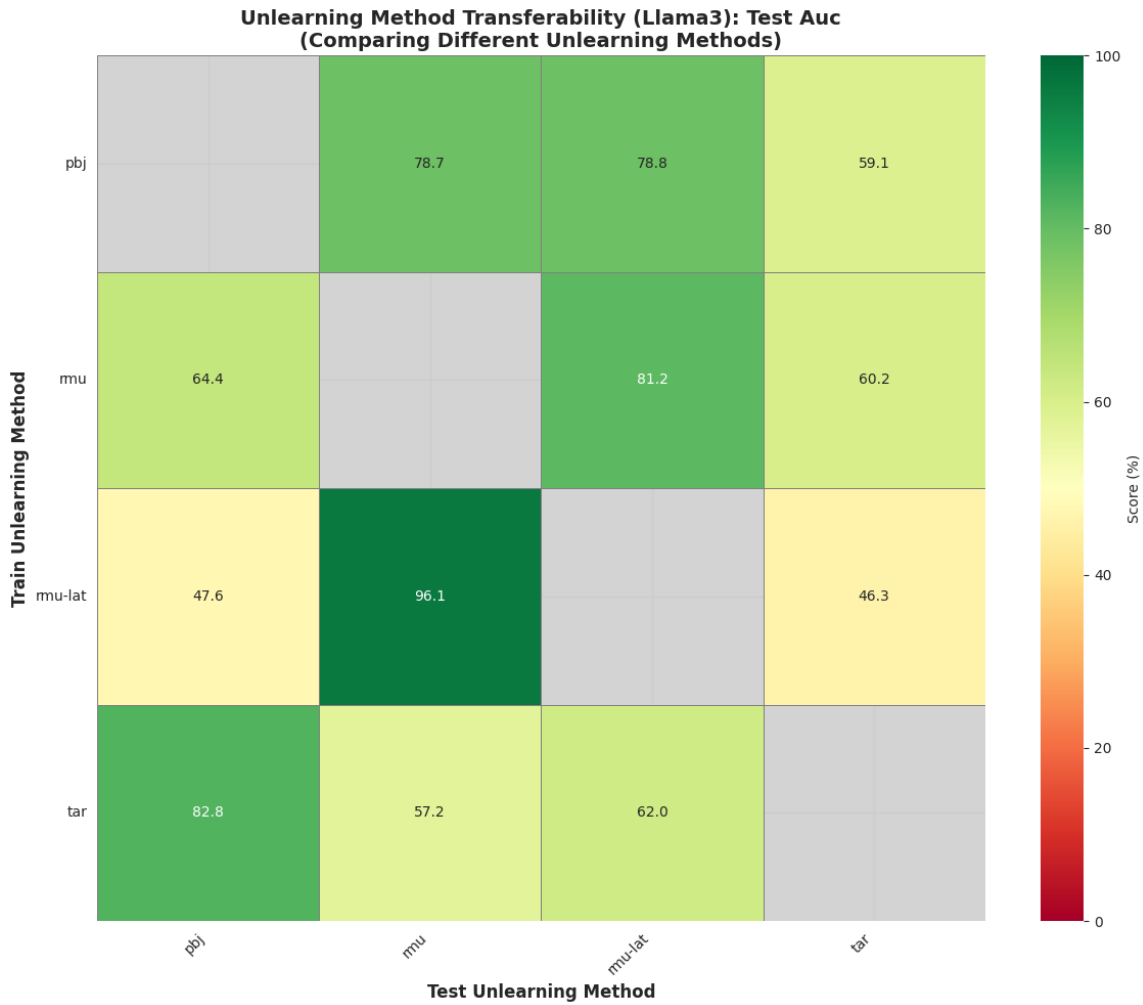


Figure 39: **Unlearning Method Transferability (Llama3): Test Auc.** This heatmap explicitly evaluates tests where classifiers trained over one unlearning method are assessed on a model unlearned via a different method (both over Llama3 base model and same dataset). Asymmetrical performances underscore the differences in geometric loss landscape specializations, such as *rmu-lat*'s exceptionally specialized capability of identifying *rmu* behavior without a symmetrical counterpart.

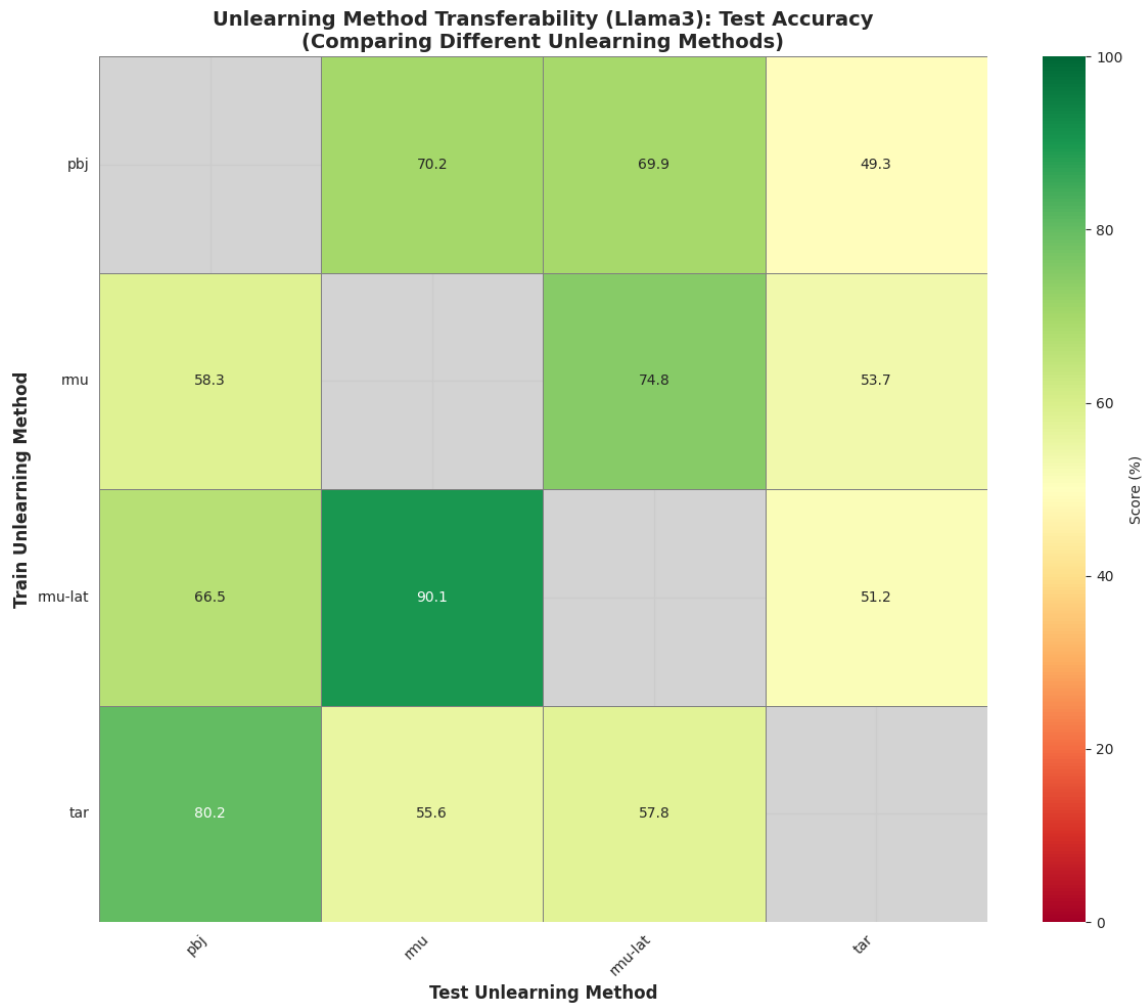


Figure 40: **Unlearning method classification accuracy heatmap.** Classification accuracy for different unlearning methods, showing performance consistency across methods. Grey cells indicate scenarios that were not tested due to data unavailability.

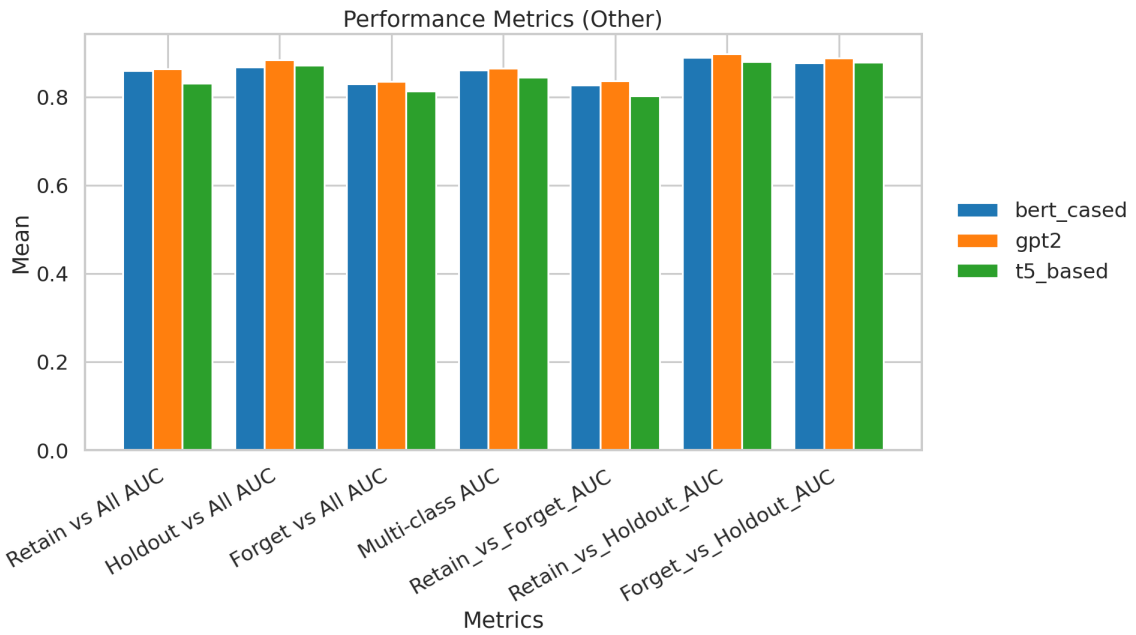


Figure 41: **Tokenizer robustness across broad metrics.** Mean and standard deviation of various classification AUCs across different tokenizers.

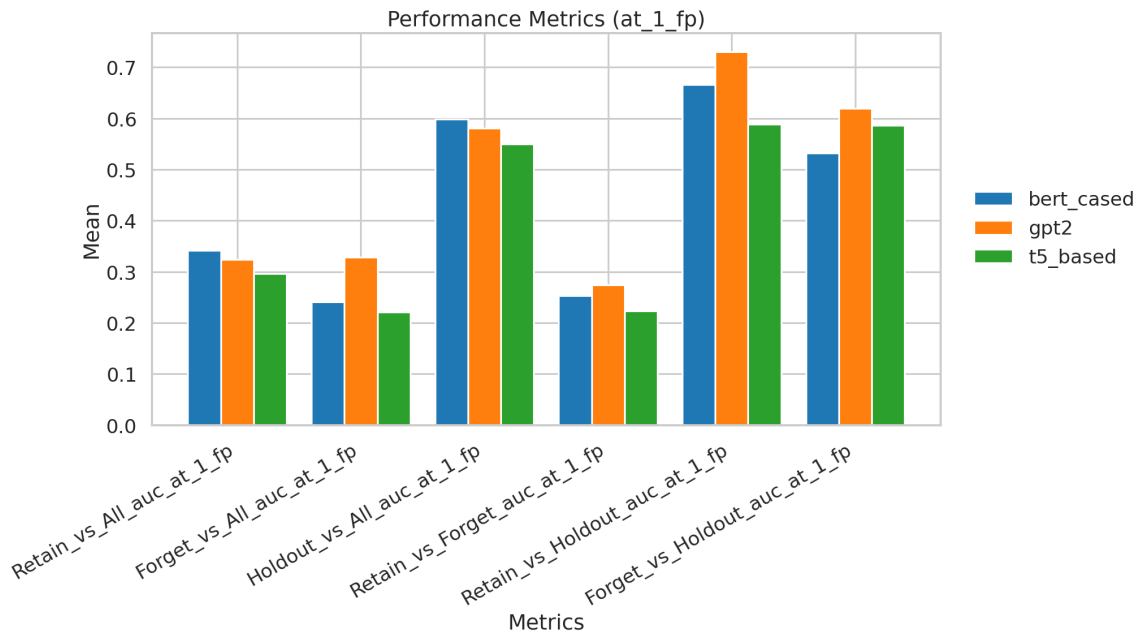


Figure 42: **Tokenizer robustness at 1% false positive rate.** Mean and standard deviation of rigorous (at 1% FPR) AUC metrics across the evaluated tokenizers.

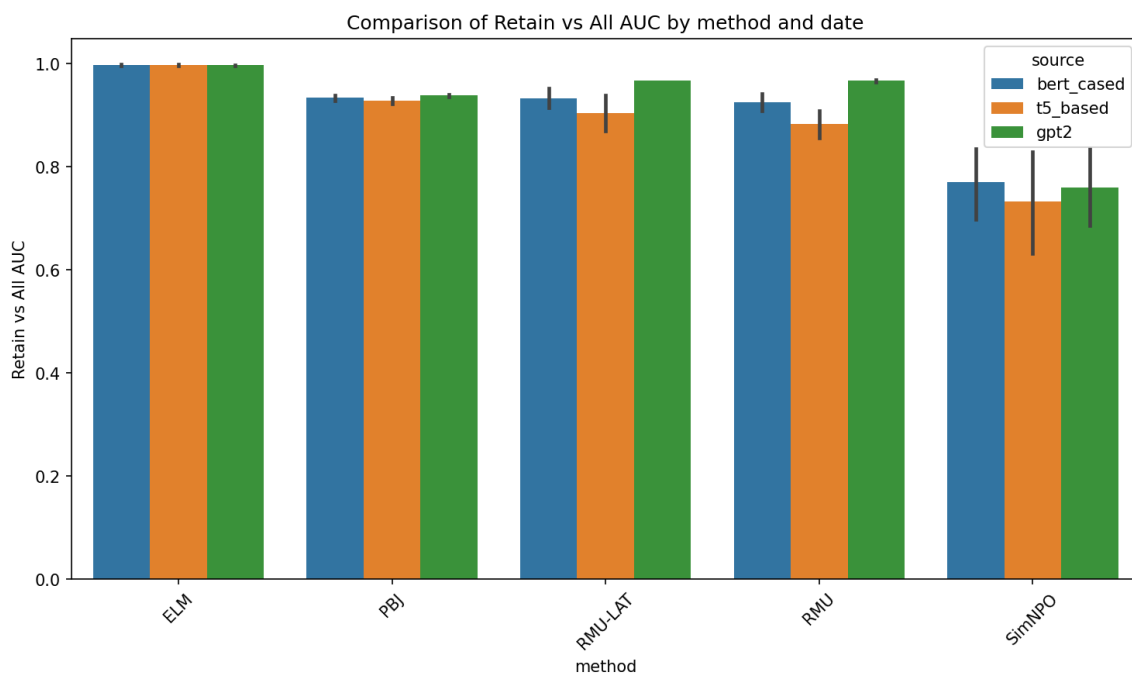


Figure 43: **Retain vs. All AUC across different tokenizers.** A method-level benchmark highlights that robust perturbation techniques consistently extract meaningful geometric features regardless of the tokenizer.

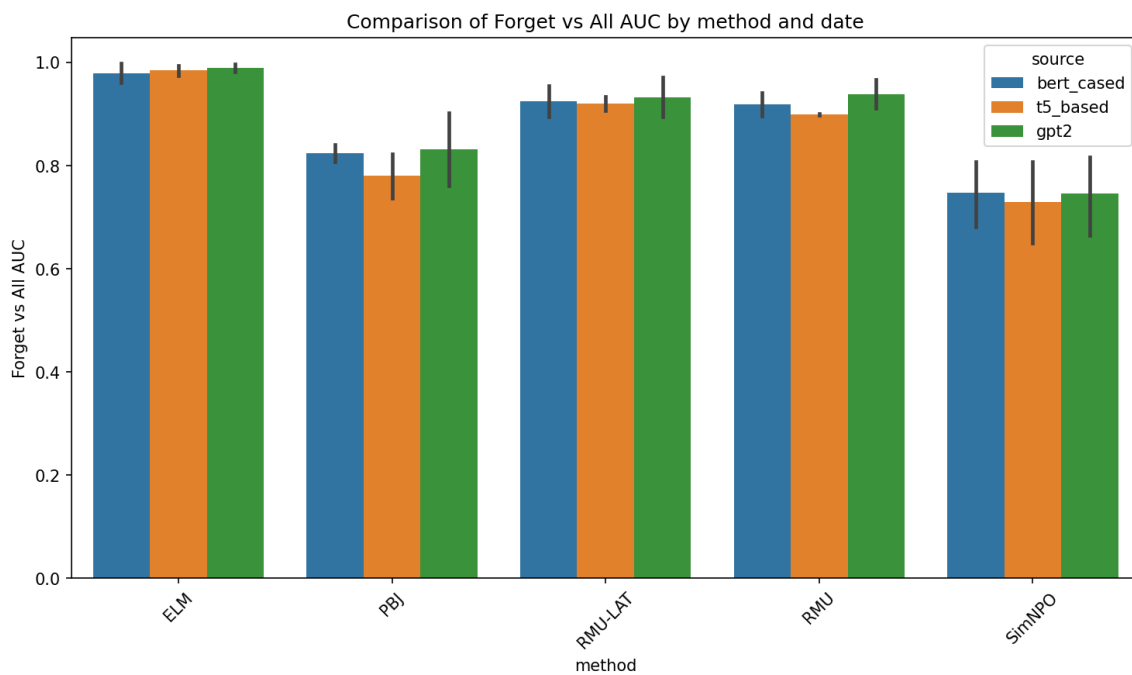


Figure 44: **Forget vs. All AUC across different tokenizers.** A method-level benchmark demonstrating the consistent distinguishability of the forget set regardless of the tokenizer.

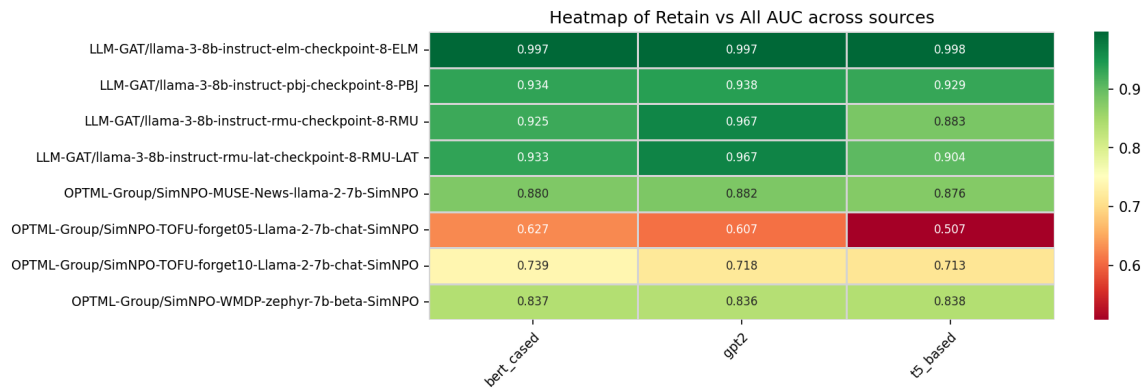


Figure 45: **Heatmap of Retain vs. All AUC.** Shows granular performance for each specific unlearning model and base configuration across different tokenizers.

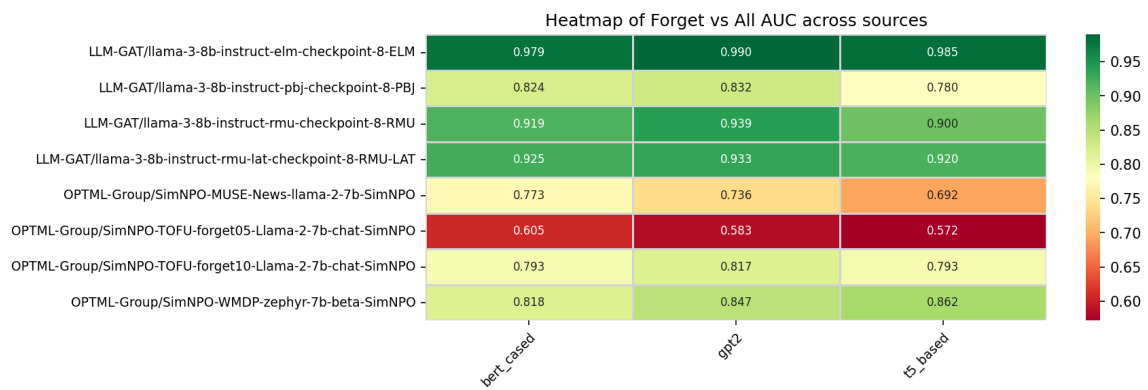


Figure 46: **Heatmap of Forget vs. All AUC.** Breaks down forget set distinguishability by tokenizer source.

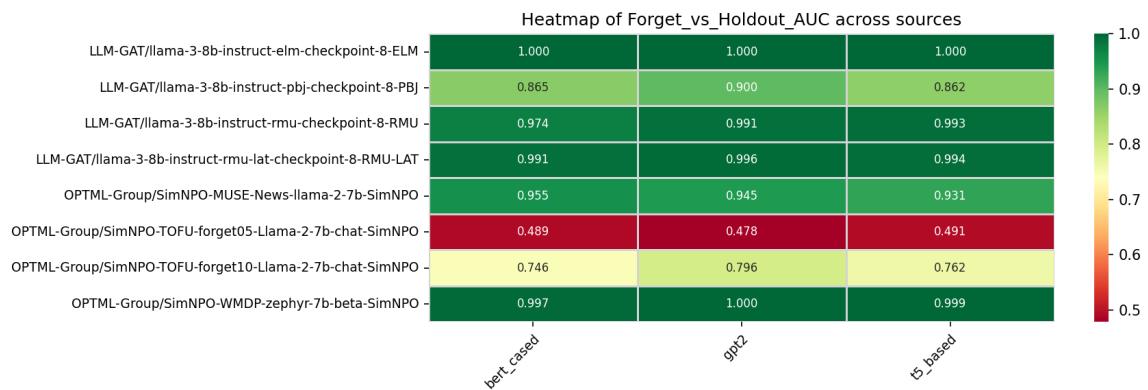


Figure 47: **Heatmap of Forget vs. Holdout AUC.** Shows granular performance for discriminating forget from holdout examples across different tokenizers.