ATTNDIFF: Attention-based Differential Fingerprinting for Large Language Models

Haobo Zhang[2,3]* **Zhenhua Xu**[1,3]* **Junxian Li**[4]
Shangfeng Sheng[5] **Dezhang Kong**[1,3]† **Meng Han**[1,3,6]†
[1]Zhejiang University [2]Renmin University of China
[3]Binjiang Institute of Zhejiang University
[4]Shanghai Jiao Tong University
[5]University of Science and Technology of China [6]Gentel.io
haobo0119@gmail.com {xuzhenhua0326, mhan}@zju.edu.cn

Abstract

Protecting the intellectual property of open-weight large language models (LLMs) requires verifying whether a suspect model is derived from a victim model despite common laundering operations such as fine-tuning (including PPO/DPO), pruning/compression, and model merging. We propose ATTNDIFF, a data-efficient **white-box** framework that extracts fingerprints from models via intrinsic information-routing behavior. ATTNDIFF probes minimally edited prompt pairs that induce controlled semantic conflicts, captures *differential attention* patterns, summarizes them with compact spectral descriptors, and compares models using CKA. Across Llama-2/3 and Qwen2.5 (3B–14B) and additional open-source families, it yields high similarity for related derivatives while separating unrelated model families (e.g., > 0.98 vs. < 0.22 with $M = 60$ probes). With 5–60 multi-domain probes, it supports practical provenance verification and accountability; our open-source implementation is available at `https://github.com/zhb0119/AttnDiff`.

1 Introduction

Large Language Models (LLMs), including Llama [44], Qwen [3], and DeepSeek [27], require substantial data and computation to develop, making their weights valuable intellectual property (IP). Open-weight releases face risks of unauthorized redistribution and model laundering [55, 65]. Model provenance verification therefore seeks to determine whether a suspect derives from a given reference model despite subsequent modification [55].

Figure 1 illustrates this problem: released checkpoints may undergo fine-tuning, pruning, merging, distillation, or quantization before appearing as suspects with uncertain origins. Verification requires fingerprints that retain evidence of lineage after these transformations.

Ownership verification methods include invasive and non-invasive approaches [55]. Invasive fingerprints embed signals through training-time modifications, including backdoor watermarks [54, 49, 24, 33]. They alter model weights, creating trade-offs between fingerprint capacity and utility and potentially introducing vulnerabilities [68], spurious triggers, or degraded generalization [50, 54, 49]. They also cannot retrospectively protect a checkpoint stolen before signal embedding.

Non-invasive, or *intrinsic*, fingerprints extract signatures without modifying parameters. Four families have distinct robustness gaps: **parameter-based** fingerprints [64] are fragile under structural

*Equal contribution.
†Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

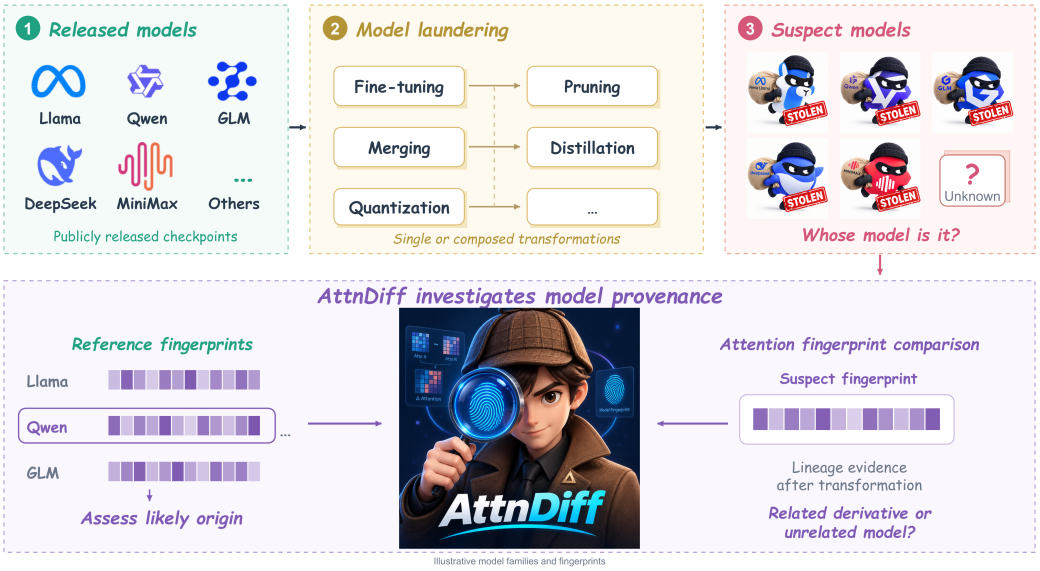


Figure 1: Model provenance verification with ATTNDIFF. **Top:** publicly released checkpoints may undergo individual or composed transformations, resulting in suspect models whose origins are uncertain. **Bottom:** ATTNDIFF independently extracts differential attention fingerprints from reference and suspect models using the same probe pairs, then compares them to assess evidence of shared lineage. Multiple references illustrate repeated pairwise comparisons. Model families, fingerprint patterns, and the highlighted match are illustrative.

edits (Sec. 5.4); **representation-based** fingerprints [61, 65] can drift under architecture changes, distillation (Appendix F.5), and preference optimization (Appendix F.1); **adversarial-example-based** fingerprints [20] rely on potentially conspicuous triggers that input perturbations can neutralize [54, 50]; and **semantic** fingerprints [35] depend on generated text and decoding, making them sensitive to paraphrasing and rewriting (Appendix F.8).

We propose ATTNDIFF, a data-efficient white-box framework based on *differential attention dynamics* under controlled semantic conflicts. Its central hypothesis is that attention re-routing encodes lineage-specific structure that persists under model modification. As illustrated in the lower panel of Figure 1, independently extracted reference and suspect fingerprints provide evidence for assessing shared lineage; comparisons can be repeated across candidate references. Concretely, single-word pivots create minimally edited prompt pairs, such as "parallel lines never intersect" versus "always intersect". ATTNDIFF extracts their attention differences across layers and heads, compresses these into spectral descriptors, and compares models using centered linear CKA [21]. This comparison captures inter-probe structure while allowing different feature dimensions. Our experiments assess whether these fingerprints remain similar for related derivatives while separating unrelated models under common transformations.

2 Related work

2.1 Invasive fingerprints

Invasive fingerprints embed ownership signals by modifying models during (pre-)training, typically via weight watermarks or backdoor triggers. Weight watermark-based schemes encode verifiable patterns in parameter space [67, 15, 10, 28, 41] but face capacity–utility trade-offs and may lose robustness under pruning, quantization, or strong fine-tuning [55]. Backdoor-based fingerprints use trigger datasets for black-box attribution [49, 5, 57, 54, 51, 56, 25, 40], yet require training-time control and can be removed by targeted erasure [66]. Recent work explores scalable fingerprint transfer [53, 52], harmless editing [63], and adversarial robustness [47, 69].

2.2 Intrinsic fingerprints

Non-invasive (intrinsic) fingerprints do not inject external signals and instead mine model-inherent properties for provenance verification [55]. Representative families include parameter-space statistics in weight space [64], representation-based descriptors over hidden states/logits [65, 61, 43, 70], semantic output traces [35, 48], and adversarial-example-based triggers [20, 14]. However, these signals can be brittle under realistic transformations, e.g., pruning/architecture edits, distillation, or preference optimization, and/or sensitive to decoding and input perturbations, limiting robustness across fine-tuning, pruning/merging, and distillation.

Among intrinsic approaches, ATTNDIFF departs from parameter-, representation-, and output-level signatures by fingerprinting models through their *differential attention* responses to controlled semantic conflicts, with similarity measured via CKA.

3 Threat model

We study a post-hoc model forensics setting where a *defender* (the legitimate model owner) verifies the provenance of a released or seized *suspect* model, while an *attacker* seeks to evade attribution after illicitly obtaining the model weights. We focus on realistic **white-box** model laundering operations that modify a stolen model's parameters and/or architecture while maintaining utility.

Attacker capabilities. With full white-box access, the attacker can apply arbitrary transformations, including supervised fine-tuning (SFT), alignment and PO (e.g., reinforcement learning from human feedback (RLHF) with Proximal Policy Optimization (PPO) or Direct Preference Optimization (DPO)), pruning, quantization, distillation to a different architecture, and model merging or weight interpolation. This setting is consistent with prior work on model fingerprinting [60, 20], where attackers are assumed to have access to the original model and can fine-tune it using validation data to evade attribution.

We also evaluate an adaptive attacker who knows the extraction procedure and obtains some defender probes, then trains the suspect to suppress differential attention (Sec. 5.7).

Defender capabilities. The defender has access to the original reference model and is granted white-box access to the suspect model during verification, consistent with legal/auditing workflows. This allows the defender to inspect internal states and compute fingerprints for both models under the same probing protocol. Verification requires inference-time attention access, but no gradients or model updates.

4 Design of ATTNDIFF

4.1 Motivation

In post-hoc LLM forensics, a suspect is rarely an "exact-copy": stolen checkpoints are routinely laundered through fine-tuning/alignment, pruning, merging, or distillation. A practical fingerprint should therefore remain stable under such transformations, where weight-space statistics can break under structural edits and hidden-state/logit signatures can drift after strong adaptation.

We encountered an empirical turning point while stress-testing small, highly controlled probes. We constructed minimally edited origin/corrupted prompt pairs that preserve surface form while flipping semantics via a single-word lexical pivot. Under these "controlled semantic conflicts", attention did not drift arbitrarily; models re-allocated attention in a structured manner. More importantly, this re-allocation appeared family-consistent: Llama-2-7B [44] derivatives exhibited closely matched profiles, whereas an unrelated model family such as Qwen2.5-7B [36] followed a systematically different regime. Together, these observations suggested that a model's intrinsic information-routing strategy—how it redistributes self-attention when resolving a semantic contradiction—may encode lineage-specific structure that survives common "laundering" operations. A detailed pilot study quantifying these re-allocation patterns via interpretable structural statistics (concentration, entropy, locality) is presented in Appendix E.

These observations motivate ATTNDIFF: by probing a model with minimally edited prompt pairs that induce controlled semantic conflicts, we can elicit a characteristic re-routing response that is

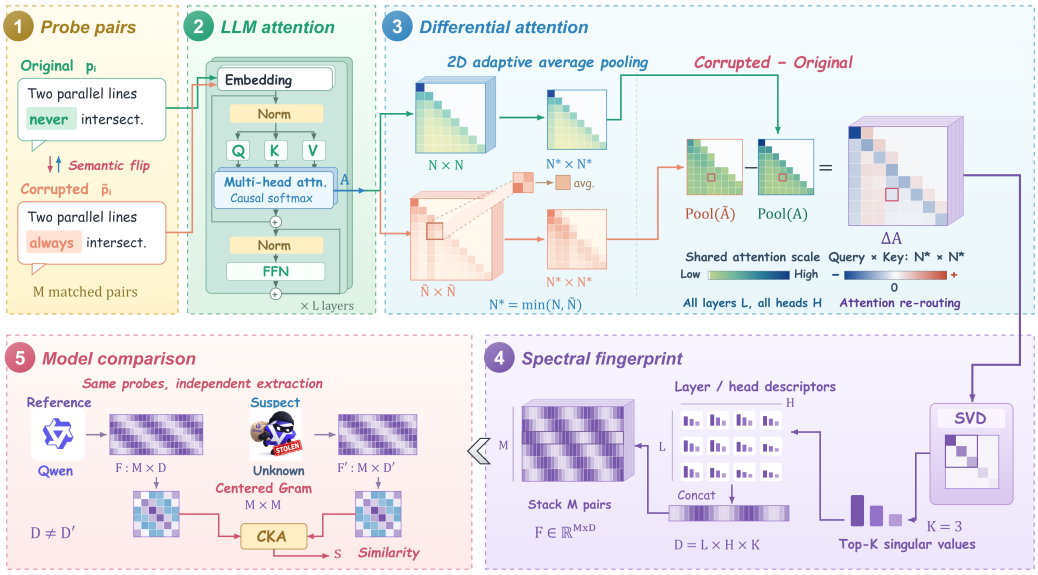


Figure 2: ATTNDIFF workflow, following steps **1–3** across the top, then **4** at bottom right and **5** at bottom left. A single-word pivot creates a semantic conflict. Causal attention maps are aligned to $N^* = \min(N, \tilde{N})$ before computing $\Delta A = \text{Pool}(\tilde{A}) - \text{Pool}(A)$. SVD yields the top-$K$ singular values per layer/head ($K = 3$), which are concatenated and stacked into $F \in \mathbb{R}^{M \times D}$, $D = LHK$. Reference and suspect fingerprints use the same ordered probes and are extracted independently. Their centered $M \times M$ Gram matrices support CKA comparison even when $D \neq D'$. Qwen is an illustrative reference model; heatmaps and spectra are schematic.

stable within a model lineage yet distinct across unrelated families. The next subsection describes how we operationalize this idea into a practical fingerprinting pipeline.

4.2 ATTNDIFF workflow

Notation. We denote a model with L layers and H attention heads per layer. For a prompt p tokenized into N tokens, $A^{(i)}_{l,h} \in \mathbb{R}^{N \times N}$ represents the post-softmax causal self-attention probability matrix at layer l and head h for the i-th probe pair. We use $\tilde{p}$ and $\tilde{A}$ to denote the corrupted prompt and its corresponding attention map. The differential attention matrix is $\Delta A = \tilde{A} - A$. The fingerprint matrix $F \in \mathbb{R}^{M \times D}$ stacks spectral descriptors from M probe pairs, where $D = LHK$ and K is the spectral rank.

ATTNDIFF fingerprints LLMs by capturing their **differential attention dynamics** under semantic conflict. Figure 2 presents five steps: (1) **Probe pairs:** construct M origin/corrupted pairs via single-word pivot substitutions; (2) **LLM attention:** extract post-softmax causal attention maps for each prompt; (3) **Differential attention:** align each pair of maps by adaptive average pooling when needed, then subtract origin from corrupted attention; (4) **Spectral fingerprint:** retain the top-K singular values per layer/head ($K = 3$ by default), concatenate the descriptors, and stack all pairs into $F \in \mathbb{R}^{M \times D}$; (5) **Model comparison:** independently extract F and F' from the reference (victim) and suspect models using the same ordered probe pairs, then compare their centered Gram matrices using CKA. The score $s = \text{CKA}(F, F')$ measures similarity in differential routing behavior. Implementation details are provided in Appendix B.

4.2.1 Probe dataset

We construct a probe set of M prompts spanning six domains: *Code*, *Math*, *Economics*, *Medicine*, *Daily QA*, and *Safe Alignment*. Since post-training typically specializes a model toward a particular domain, we diversify probe domains to reduce domain-specific drift in internal routing behaviors and to provide complementary anchors for fingerprint comparison. In addition, *Safe Alignment*

is included to cover policy-gated behaviors (e.g., refusal or cautious responses) that can follow a distinct internal routing regime compared to capability-oriented queries, providing a complementary anchor under realistic instruction/alignment settings.

We then construct M origin/corrupted probe pairs for fingerprint extraction, as detailed below.

4.2.2 Differential fingerprint

For the i-th probe pair $(p_i, \tilde{p}_i)$, we generate a corrupted prompt $\tilde{p}_i$ from the origin prompt p_i via a *single-word (lexical) pivot* substitution, yielding a *semantic conflict*. Tokenization is performed using the target model's tokenizer on *only the prompt string itself* (excluding chat templates or system prefixes). Typically $\tilde{N} = N$, with rare mismatches due to tokenizer segmentation.

For each layer and head, we extract the post-softmax causal self-attention probability matrices $A^{(i)}_{l,h} \in \mathbb{R}^{N\times N}$ and $\tilde{A}^{(i)}_{l,h} \in \mathbb{R}^{\tilde{N}\times\tilde{N}}$, where strictly upper-triangular entries are masked to zero. We refer to the stacked tensors as the **origin attention map** $\mathcal{A}^{(i)} \in \mathbb{R}^{L\times H\times N\times N}$ and the **corrupted attention map** $\tilde{\mathcal{A}}^{(i)} \in \mathbb{R}^{L\times H\times\tilde{N}\times\tilde{N}}$.

To handle the rare case $\tilde{N} \neq N$, we align attention maps to a common resolution $N^* = \min(N, \tilde{N})$ via **2D adaptive average pooling**, as illustrated in Figure 3. This operator partitions the input matrix into bins and averages within each bin to produce the target resolution, preserving coarse-grained attention distribution without introducing boundary artifacts. We apply this operator to pool each $A^{(i)}_{l,h}$ and $\tilde{A}^{(i)}_{l,h}$ to $\bar{A}^{(i)}_{l,h}, \bar{\tilde{A}}^{(i)}_{l,h} \in \mathbb{R}^{N^*\times N^*}$.

We then compute the **differential attention matrix** $\Delta A^{(i)}_{l,h} = \bar{\tilde{A}}^{(i)}_{l,h} - \bar{A}^{(i)}_{l,h}$ (i.e., $\mathrm{Pool}(\tilde{A}^{(i)}_{l,h}) - \mathrm{Pool}(A^{(i)}_{l,h})$), which highlights attention re-routing induced by semantic conflict under aligned token topology. Step 3 in Figure 2 uses a shared scale for origin/corrupted attention and a zero-centered diverging scale for their signed difference. To obtain a compact, length-invariant signature, we apply truncated SVD to $\Delta A^{(i)}_{l,h}$ and retain the top-K singular values as a spectral descriptor $\mathbf{s}^{(i)}_{l,h} \in \mathbb{R}^K$. Step 4 labels this operation "Top-K singular values" and illustrates the default $K = 3$. Concatenating across all layers and heads yields a one-dimensional instance-level fingerprint vector $\mathbf{x}^{(i)} \in \mathbb{R}^{LHK}$, and stacking over M probes forms the fingerprint matrix $F \in \mathbb{R}^{M\times LHK}$.

4.2.3 Similarity measurement by CKA

To compare two models, we independently compute fingerprints F and F' on the same ordered probe pairs and measure similarity using **CKA**, as shown in step 5 of Figure 2. Specifically, we form linear Gram matrices $K = FF^\top$ and $K' = F'F'^\top \in \mathbb{R}^{M\times M}$ that encode inter-probe relational geometry. Let $H = I - \frac{1}{M}\mathbf{1}\mathbf{1}^\top$ be the centering matrix and define centered Gram matrices $\bar{K} = HKH$ and $\bar{K}' = HK'H$. We compute CKA via the Hilbert-Schmidt Independence Criterion (HSIC), where $\mathrm{HSIC}(K, K') \propto \mathrm{tr}(\bar{K}\,\bar{K}')$: $\mathrm{CKA}(F, F') = \frac{\mathrm{HSIC}(K,K')}{\sqrt{\mathrm{HSIC}(K,K)\mathrm{HSIC}(K',K')}}$. We use centered Gram matrices in HSIC to remove mean effects and focus the comparison on similarity structure across probes.

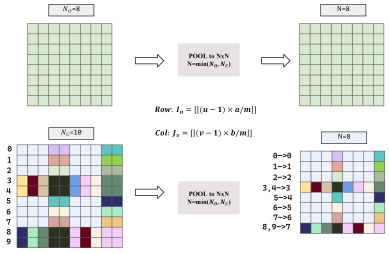


Figure 3: Attention map alignment via 2D adaptive average pooling to resolution $N^* = \min(N, \tilde{N})$. This preserves coarse-grained routing structure without boundary artifacts when tokenization induces length mismatch.

Critically, practical transformations such as structured pruning, distillation, or architecture edits may change model structure (e.g., removing layers or attention heads), yielding fingerprints with different feature dimensions (e.g., $F \in \mathbb{R}^{M\times D}$ and $F' \in \mathbb{R}^{M\times D'}$ with $D \neq D'$ due to different L or H). This makes direct feature matching ill-posed. In contrast, CKA operates on probe-indexed Gram matrices in $\mathbb{R}^{M\times M}$ and requires matching probe identities and ordering, enabling direct comparison without explicit layer/head alignment. Consequently, ATTNDIFF remains applicable across models

with heterogeneous depths and head counts, while retaining robustness to re-parameterizations and other real-world modifications.

5 Experiment

We compare ATTNDIFF against non-invasive baselines under fine-tuning, merging, and pruning (Sec. 5.2–5.4), followed by ablations (Sec. 5.5), discriminability analysis (Sec. 5.6), and adaptive suppression (Sec. 5.7). Figure 4 summarizes robustness across deployment transformations, including preference optimization (Appendix F.1) and distillation (Appendix F.5).

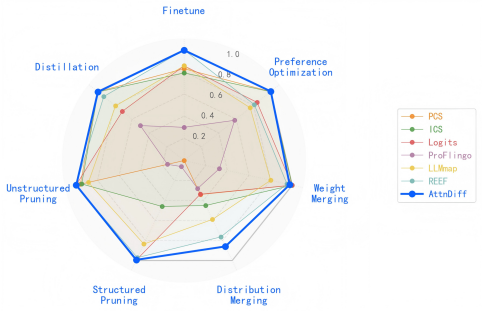


Figure 4: Radar comparison across deployment transformations. Larger radii indicate stronger robustness.

5.1 Experimental setting

Probe set. Unless otherwise specified, we construct a probe set with $M = 60$ prompts spanning six domains, with 10 prompts per domain: *Code*, *Math*, *Economics*, *Medicine*, *Daily QA*, and *Safe Alignment*. For each origin prompt p, we generate a minimally edited corrupted prompt $\tilde{p}$ via a single-word pivot and compute fingerprints on the resulting M probe pairs.

Model. We evaluate ATTNDIFF under realistic model transformation scenarios, covering diverse base architectures and a broad suite of derivative models. Our experiments cover models from the Llama-2/3 [44, 13], Qwen2.5 [36], Gemma [42], and Mistral [18] families, spanning parameter scales from 1B to 14B and including downstream adaptations and optimizations such as supervised fine-tuning, preference optimization, knowledge distillation, pruning, and model merging.

Baselines. Following the taxonomy in the Introduction, we select representative *non-invasive* fingerprinting baselines that cover both black-box and white-box access assumptions. Specifically, we include PCS/ICS [64] as *parameter-based* fingerprints, Logits [61] and REEF [65] as *representation-based* fingerprints, ProFlingo [20] as an *adversarial-example-based* fingerprint, and LLMMap [35] as a *semantic* fingerprint. For each method, we report the similarity score produced by its original formulation.

Our experiments ask: **RQ1:** How robust is ATTNDIFF to fine-tuning and preference optimization? **RQ2:** Does it withstand diverse merging strategies? **RQ3:** How robust is it to pruning/compression? **RQ4:** Does probe-aware suppression transfer to held-out or refreshed probes?

5.2 Model fine-tuning

Table 1: SFT robustness results (similarity score) on Llama-2-7B-derived suspect models.

	Llama-2-Finance-7B (5M tokens)	Vicuna-1.5-7B (370M tokens)	WizardMath-7B (1.8B tokens)	ChineseLLaMA-2-7B (13B tokens)	CodeLLaMA-7B (500B tokens)	Llemma-7B (700B tokens)	Avg
PCS	0.9979	0.9985	0.9965	0.9390	0.5301	0.5052	0.8279
ICS	0.9952	0.9949	0.9985	0.7309	0.5112	0.5104	0.7902
Logits	0.9999	0.9999	0.9999	0.7033	0.7833	0.6367	0.8538
REEF	0.9950	0.9985	0.9979	0.9974	0.9947	0.9962	0.9966
ProFlingo	0.2400	0.5200	0.4200	0.2800	0.2000	0.1400	0.3000
LLMMap	0.8986	0.7294	0.7691	0.8720	0.9555	0.8998	0.8541
Ours	0.9989	0.9986	0.9985	0.9963	0.9890	0.9856	0.9945

Suspect model headers include the fine-tuning data scale. Cell shading (per column): =best, =second, =third.

Fine-tuning is a critical stage in the LLM lifecycle for adapting base models to downstream tasks or aligning them with human values [34, 37]. This process involves extensive parameter updates that can significantly shift the model's internal representations and output distributions, posing a severe test for fingerprint robustness [32].

For SFT, we use Llama-2-7B as the victim model and collect suspect models fine-tuned with different data scales and application domains, including Llama-2-finance-7b [31], Vicuna-1.5-7b [6], WizardMath-7b [29], Chinese-LLaMA-2-7b [7], CodeLLaMA-7b [39], and Llemma-7b [2].

Table 1 indicates that large-scale SFT can substantially alter model parameters/representations, causing parameter-based fingerprints (PCS/ICS) to drop sharply on heavily fine-tuned suspects. Logits also drops to 0.7833/0.6367 on CodeLLaMA-7B and Llemma-7B. In contrast, ATTNDIFF maintains uniformly high similarity (> 0.985) across all SFT suspects and is comparable to the SOTA baseline REEF.

5.3 Model merge

Table 2: Model merge robustness results (similarity score) on open-source merged suspects.

	Weight Merging (Evollm-jp-7b)			Distribution Merging (Fusellm-7b)		
	Shisa-gamma-7b-v1	Wizardmath-7b-1.1	Abel-7b-002	Llama-2-7b	Openllama-2-7b	Mpt-7b
PCS	0.9992	0.9990	0.9989	0.9997	0.0194	0.0000
ICS	0.9992	0.9988	0.9988	0.9986	0.2478	0.1014
Logits	0.9933	0.9999	0.9999	0.9999	0.0100	0.0000
REEF	0.9635	0.9526	0.9374	0.9996	0.6713	0.6200
ProFlingo	0.3000	0.4000	0.2800	0.5400	0.1400	0.1600
LLMMap	0.7651	0.8343	0.8011	0.9511	0.5742	0.2413
Ours	0.9726	0.9561	0.9996	0.9962	0.7953	0.7851

Cell shading (per column): =*best,* =*second,* =*third.*

Model merging combines multiple pretrained models in weight space or at the distribution/behavior level, often integrating complementary capabilities without accessing training data or retraining [59, 1, 45, 46, 62, 17]. Because a merged suspect is derived from multiple victim models, its fingerprints can be mixed and harder to attribute to all sources.

From Table 2, ATTNDIFF consistently attains high similarity with all parent models in weight-space merges (≥ 0.95) and maintains substantial similarity (≈ 0.78–0.80) with heterogeneous distribution-level merges, whereas several baselines either fail to attribute all sources or exhibit noticeably weaker alignment on cross-architecture merges.

5.4 Model pruning

Model pruning compresses LLMs by removing redundant parameters to improve efficiency, often followed by retraining to recover capabilities. This poses a distinct challenge to model fingerprinting, as it fundamentally alters the model's weights and architecture. Recent studies have shown that pruning can weaken the verification effectiveness of multiple fingerprinting schemes [54, 50]. We therefore evaluate robustness against a spectrum of pruning methodologies, including both structured and unstructured pruning on open-source checkpoints as well as additional suspects generated via the LLMPruner toolkit [30].

Table 3 shows that ATTNDIFF preserves high similarity across all structured and unstructured pruned suspects (no lower than 0.9879), even when aggressive structured pruning causes parameter-based fingerprints (PCS/ICS) to collapse and ProFlingo/LLMMap to degrade substantially. These results demonstrate that our differential attention fingerprint is robust to diverse pruning and compression strategies.

5.5 Ablation study

We conduct ablation studies to validate key design choices of ATTNDIFF. Beyond the core experiments reported below, we perform extensive hyperparameter and design ablations in the appendix.

Table 3: Robustness results (similarity score) on pruned suspect models, comparing structured and unstructured pruning strategies.

	Structured Pruning						Unstructured Pruning		
	Sheared-llama-1.3b-pruned	Sheared-llama-1.3b	Sheared-llama-1.3b-sharegpt	Sheared-llama-2.7b-pruned	Sheared-llama-2.7b	Sheared-llama-2.7b-sharegpt	Sparse-llama-2-7b	Wanda-llama-2-7b	GBLM-llama-2-7b
PCS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9560	0.9620	0.9616
ICS	0.4927	0.3512	0.3510	0.6055	0.4580	0.4548	0.9468	0.9468	0.9478
Logits	0.9967	0.9999	0.9999	0.9967	0.9999	0.9999	0.9999	0.9999	0.9999
REEF	0.9368	0.9676	0.9710	0.9278	0.9701	0.9991	0.9985	0.9986	0.9991
ProFlingo	0.0400	0.0200	0.0800	0.0200	0.1000	0.0800	0.1600	0.1200	0.1800
LLMMap	0.9088	0.9007	0.8152	0.9400	0.9236	0.7072	0.8459	0.9145	0.8956
Ours	0.9879	0.9938	0.9903	0.9929	0.9952	0.9936	0.9996	0.9927	0.9995

Cell shading (per column): =*best,* =*second,* =*third.*

Differential mechanism (Table 12): comparing against a non-differential baseline that uses standard attention maps, the differential design is crucial under heavy domain adaptation—CodeLlama-7b and Llemma-7b drop to $0.51/0.29$ without differencing but recover to $0.99/0.99$ with ATTNDIFF ($\Delta > 0.47$). For unrelated models, differencing reduces spurious similarity from ~ 0.4 to < 0.23.

Probe configuration (Figure 6, Table 13): sparse probing ($M \leq 15$) yields high spurious similarity among unrelated models; $M = 60$ provides optimal discrimination. Domain-specific fine-tuning induces localized shifts in expert domains, but aggregated fingerprints remain stable for related suspects (> 0.98) and well separated from unrelated architectures (< 0.22).

Probe length (Appendix B.1, Table 10): medium-length prompts ($\ell \approx 30$ words) balance semantic context and attention stability.

Spectral rank (Appendix B.2, Table 11): $K = 3$ optimally trades off discriminability and compactness.

Similarity metric (Appendix B.3, Table 15): CKA achieves superior discriminability over CCA/SVCCA with lower spurious similarity on unrelated models (0.1647 vs 0.3112/0.2419).

5.6 Discriminability analysis

While Sec. 5.2–5.4 demonstrate that ATTNDIFF maintains high similarity for related derivatives, practical fingerprinting also requires strong *discriminability*: the ability to reject unrelated models with high confidence. We therefore evaluate ATTNDIFF as a binary classifier that labels each suspect as either a derivative (positive) or unrelated (negative), and report both fixed-threshold and threshold-free discriminability metrics.

Table 4: Threshold-free discriminability (AUROC/AUPR, related=positive).

Method	AUROC	AUPR
PCS	0.9566	0.9367
ICS	0.9789	0.9795
Logits	0.9757	0.9746
REEF	0.9867	0.9877
ProFlingo	0.8374	0.8927
LLMMap	0.8925	0.9014
Ours	**0.9883**	**0.9881**

ATTNDIFF *attains highest AUROC/AUPR, indicating strong separability.*

Experimental setup. We construct two model pools: (i) **Related suspects** ($n = 36$, positive class): Llama-2-7B derivatives via fine-tuning, PPO/DPO, pruning/compression, and merging (Tables 1, 3, A1, A2, A3); (ii) **Unrelated suspects** ($n = 33$, negative class): distinct architectures/families from cross-family evaluation and ablations (Tables 12, A3, A4, A5).

For a threshold t, we predict *positive* (derivative) if $\mathrm{CKA}(F, F') \geq t$, otherwise *negative* (unrelated). We report two types of metrics: (i) **Fixed-threshold**—Precision (P), Recall (R), and F1-score (F1) at $t \in \{0.10, 0.25, 0.50, 0.90\}$; (ii) **Threshold-free**—AUROC and AUPR, treating related suspects as the positive class.

Results and analysis. ATTNDIFF achieves the highest F1 at $t = 0.25$ (0.9000) and $t = 0.90$ (0.8750). At $t = 0.25$, it is the only method with perfect recall (R=1.0000) while maintaining strong precision (P=0.8182), effectively balancing zero missed derivatives with controlled false positives.

Table 5: Discriminability at fixed thresholds $t \in \{0.10, 0.25, 0.50, 0.90\}$: precision (P), recall (R), F1-score.

Method	$t = 0.10$			$t = 0.25$			$t = 0.50$			$t = 0.90$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
PCS	0.6596	0.8611	0.7470	0.6739	0.8611	0.7561	0.8108	0.8333	0.8219	0.8788	**0.8056**	0.8406
ICS	0.5714	**1.0000**	0.7273	0.8649	0.8889	0.8767	0.9394	0.8611	0.8986	0.9630	0.7222	0.8254
Logits	0.5556	0.9722	0.7071	0.7778	0.9722	0.8642	0.8947	**0.9444**	**0.9189**	0.9655	0.7778	0.8615
REEF	0.6545	**1.0000**	**0.7912**	0.8611	0.8611	0.8611	**1.0000**	0.7778	0.8750	**1.0000**	0.6389	0.7797
ProFlingo	**0.7778**	0.7778	0.7778	**0.9524**	0.5556	0.7018	**1.0000**	0.2222	0.3636	**1.0000**	0.1389	0.2439
LLMMap	0.5217	**1.0000**	0.6857	0.6481	0.9722	0.7778	0.7143	0.8333	0.7692	0.8571	0.3333	0.4800
Ours	0.5455	**1.0000**	0.7059	0.8182	**1.0000**	**0.9000**	0.8500	**0.9444**	0.8947	**1.0000**	0.7778	**0.8750**

ATTNDIFF *achieves best F1 at* $t = 0.25$ *(0.9000).*

At $t = 0.90$, it achieves perfect precision (P=1.0000, FP=0) with the highest F1, demonstrating robust separability across practical operating points.

Table 4 further confirms this via threshold-free metrics: ATTNDIFF attains the highest AUROC (0.9883) and AUPR (0.9881), outperforming all baselines. Notably, REEF is competitive (0.9867/0.9877), while ProFlingo and LLMMap drop to ≈ 0.83–0.90, reflecting their sensitivity to surface-level transformations. Together, these results demonstrate that ATTNDIFF provides a practical and discriminative tool for binary provenance decisions across realistic operating points.

5.7 Adaptive suppression and probe refresh

Setup. We evaluate probe-aware suppression on a related Gemma-2 model pair under a fixed 96-step LoRA budget, with an ordinary-fine-tuning control. The attacker observes 20–80% of a 60-pair attack pool; evaluation compares leaked probes, held-out probes from that pool, and a disjoint 60-pair refresh pool. Training details, the suppression objective, and probe allocation are provided in Appendix B.4.

Table 6: Probe-aware suppression under a fixed 96-step budget. CKA compares each adapted suspect with the frozen reference. FT denotes ordinary fine-tuning; PPL and token accuracy measure attack utility on an independent WikiText-2 slice.

Exposure	**Adaptive attack: CKA**			**FT: CKA**	**Attack utility**	
(pairs)	Leaked	Held-out	Refreshed	Refreshed	PPL ↓	Acc. ↑
20% (12/60)	0.738	0.967	0.991	0.998	21.951	0.4156
40% (24/60)	0.964	0.935	0.990	0.998	22.319	0.4096
60% (36/60)	0.934	0.964	0.988	0.998	23.053	0.4082
80% (48/60)	0.917	0.987	0.988	0.998	23.746	0.4033

Transfer and utility. At 20% exposure, leaked CKA falls from its clean value of 0.992 to 0.738. Transfer is limited: held-out CKA remains 0.935–0.987 and refreshed CKA 0.988–0.991, versus approximately 0.998 for ordinary fine-tuning; even at 80% exposure, refreshed CKA is 0.988. On an independent 8,106-token WikiText-2 slice, clean perplexity (PPL)/next-token accuracy are 27.691/0.4003; all attacked models improve both measured metrics (Table 6). Ordinary fine-tuning yields better PPL/accuracy (19.725–19.891/0.4282–0.4305), indicating an adaptation cost from suppression, although attacked models do not collapse relative to the clean suspect.

Under this model pair and fixed budget, suppression primarily fits exposed probes, supporting private, refreshed probes for verification. Minimizing attention-difference norms is not direct optimization of CKA; this pilot does not establish robustness to unrestricted adaptive attacks or preservation of all downstream capabilities.

6 Discussion

ATTNDIFF supports post-hoc provenance verification under common laundering pipelines, including fine-tuning, preference optimization, pruning/compression, and merging, while maintaining a clear separation margin from unrelated architectures.

Probe refresh. The adaptive experiment (Sec. 5.7) supports keeping a private reserve and replacing exposed probes: suppression transfers poorly to a disjoint refresh pool under the evaluated budget. Refreshing requires recomputing reference and suspect fingerprints on the same new pairs, without retraining either model.

Partial access. Dense next-token log-probabilities permit a possible gray-box extension using differential output features. Appendix C reports a small DiffLogits pilot and specifies a reconstructed evaluation protocol. This extension is separate from attention-based ATTNDIFF and does not validate text-only APIs.

Computational cost. ATTNDIFF needs no training or gradient optimization: M probe pairs require $2M$ forward passes per model with attention recording, followed by lightweight pooling (if needed), differencing, top-K SVD, and CKA. Caching the victim fingerprint leaves $2M$ suspect passes as the dominant cost. With $M = 60$, a 7B-parameter model takes approximately 5 minutes on one NVIDIA RTX 5090 GPU. This avoids tensor construction and alignment heuristics (PCS/ICS), trigger optimization (ProFlingo), and inference model training (LLMMap); Appendix D.3 provides a detailed comparison.

7 Conclusion

ATTNDIFF is a white-box post-hoc fingerprinting method that captures model-specific information routing through *differential attention* under controlled semantic conflicts. Compact spectral descriptors and centered linear CKA yield model-agnostic fingerprints that remain stable for co-originated derivatives and separated from unrelated architectures across fine-tuning (including PPO/DPO), pruning/compression, and merging. The suppression experiment supports refreshing exposed probes under the evaluated budget. Small probe sets and lightweight computation make ATTNDIFF practical for provenance verification and accountability in open-weight LLMs, although adoption risks unwarranted copyright disputes (Appendix G).

Limitations and future work

Access and evaluation scope. ATTNDIFF requires internal attention activations. The DiffLogits pilot (Appendix C) requires dense log-probabilities and covers only one related and one unrelated suspect; its reconstructed protocol awaits numerical re-evaluation. Text-only and top-k-only APIs remain outside the evidence. The adaptive attack study likewise covers one model pair and a fixed budget.

Theoretical modeling of laundering effects. While the centered linear CKA comparison provides useful invariance to feature rotations, permutations, and isotropic scaling, we still lack a complete mathematical model that propagates common laundering operations (e.g., fine-tuning, pruning, merging, and distillation) from parameter or architecture transformations to perturbations in the extracted ATTNDIFF fingerprints and the resulting similarity scores. Establishing such an end-to-end modeling framework remains an important direction for future work.

Acknowledgments

This work was supported by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (Grant No. 2026SDXT012) and the Preferential Funding for Postdoctoral Projects in Zhejiang Province (No. ZJ2026029).

Competing interests. The authors declare no competing interests.

References

[1] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7(2):195–204, 2025.

[2] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Qiaochu Jiang, Jia Deng, Stella R Biderman, and Sean Welleck. Llemma: An open language model for mathematics. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 40622–40649, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/b225f5c7cd13615e9558c3931fa4e66f-Paper-Conference.pdf.

[3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. URL https://arxiv.org/abs/2309.16609.

[4] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. URL https://arxiv.org/abs/2212.08073.

[5] Jiacheng Cai, Jiahao Yu, Yangguang Shao, and Yuhang Wu. Utf:undertrained tokens as fingerprints a novel approach to llm identification, 2025. URL https://arxiv.org/abs/2410.12318.

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

[7] Yiming Cui, Ziqing Yang, and Xin Yao. Efficient and effective text encoding for chinese llama and alpaca, 2024. URL https://arxiv.org/abs/2304.08177.

[8] MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXV*, pages 270–287, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-73225-6. doi: 10.1007/978-3-031-73226-3_16. URL https://doi.org/10.1007/978-3-031-73226-3_16.

[9] Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. Della-merging: Reducing interference in model merging through magnitude-based sampling, 2024. URL https://arxiv.org/abs/2406.11617.

[10] Pierre Fernandez, Guillaume Couairon, Teddy Furon, and Matthijs Douze. Functional invariants to watermark large transformers. *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4815–4819, 2023. URL https://api.semanticscholar.org/CorpusID:264172180.

[11] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023. URL https://arxiv.org/abs/2210.17323.

[12] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom Brown, Nicholas Joseph, Sam McCandlish, Chris Olah, Jared Kaplan, and Jack Clark. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned, 2022. URL https://arxiv.org/abs/2209.07858.

[13] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[14] Martin Gubri, Dennis Ulmer, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. TRAP: Targeted random adversarial prompt honeypot for black-box identification. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11496–11517, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.683. URL https://aclanthology.org/2024.findings-acl.683/.

[15] Qingxiao Guo, Xinjie Zhu, Yilong Ma, Hui Jin, Yunhao Wang, Weifeng Zhang, and Xiaobing Guo. Invariant-based robust weights watermark for large language models, 2025. URL https://arxiv.org/abs/2507.08288.

[16] Harold Hotelling. *Relations Between Two Sets of Variates*, pages 162–190. Springer New York, New York, NY, 1992. ISBN 978-1-4612-4380-9. doi: 10.1007/978-1-4612-4380-9_14. URL https://doi.org/10.1007/978-1-4612-4380-9_14.

[17] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2023. URL https://arxiv.org/abs/2212.04089.

[18] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, et al. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[19] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

[20] Heng Jin, Chao-Yu Zhang, Shang-Hao Shi, Wen Jing Lou, and Y. Thomas Hou. Proflingo: A fingerprinting-based intellectual property protection scheme for large language models. *2024 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9, 2024. URL https://api.semanticscholar.org/CorpusID:270738202.

[21] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

[22] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

[23] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison,

Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In Heike Adel and Shuming Shi, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-demo.21. URL https://aclanthology.org/2021.emnlp-demo.21/.

[24] Peixuan Li, Pengzhou Cheng, Fangqi Li, Wei Du, Haodong Zhao, and Gongshen Liu. Plmmark: a secure and robust black-box watermarking framework for pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14991–14999, 2023.

[25] Shen Li, Liuyi Yao, Jinyang Gao, Lan Zhang, and Yaliang Li. Double-i watermark: Protecting model copyright for llm fine-tuning, 2024. URL https://arxiv.org/abs/2402.14883.

[26] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

[27] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

[28] Xiaokun Luan, Zeming Wei, Yihao Zhang, and Meng Sun. Robust and efficient watermarking of large language models using error correction codes. *Proc. Priv. Enhancing Technol.*, 2025 (4):183–200, 2025. doi: 10.56553/POPETS-2025-0126. URL https://doi.org/10.56553/popets-2025-0126.

[29] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 49573–49609, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7c04aea54c2a60a632a47bd451cd2849-Paper-Conference.pdf.

[30] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 21702–21720. Curran Associates, Inc., 2023. doi: 10.52202/075280-0950. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/44956951349095f74492a5471128a7e0-Paper-Conference.pdf.

[31] Meta AI. Llama 2 community license agreement, 2023. URL https://ai.meta.com/llama/license/. Accessed: 2024-08-28.

[32] Anshul Nasery, Edoardo Contente, Alkin Kaz, Pramod Viswanath, and Sewoong Oh. Are robust llm fingerprints adversarially robust?, 2025. URL https://arxiv.org/abs/2509.26598.

[33] Anshul Nasery, Jonathan Hayase, Creston Brooks, Peiyao Sheng, Himanshu Tyagi, Pramod Viswanath, and Sewoong Oh. Scalable fingerprinting of large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 125116–125152. Curran Associates, Inc., 2025. doi: 10.52202/085713-4170. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b5ee4715ef8e96176ef3ccbe229b6ab9-Paper-Conference.pdf.

[34] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[35] Dario Pasquini, Evgenios M Kornaropoulos, and Giuseppe Ateniese. LLMmap: Fingerprinting for Large Language Models. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 299–318, 2025.

[36] Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bo-Wen Yu, Cheng-Yuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Hao-Ran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jian-Wei Zhang, Jian-Xin Yang, Jia-Xin Yang, Jing-Ren Zhou, Jun-Yang Lin, Kai Dang, Ke-Ming Lu, Ke-Qin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Run-Ji Lin, Tian-Hao Li, Tingyu Xia, Xing-Zhang Ren, Xuan-Cheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhen-Ru Zhang, Zi-Han Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024. URL https://api.semanticscholar.org/CorpusID:274859421.

[37] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc., 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

[38] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

[39] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024. URL https://arxiv.org/abs/2308.12950.

[40] Mark Russinovich, Yanan Cai, and Ahmed Salem. Hey, that's my model! introducing chain & hash, an llm fingerprinting technique. In *International Conference on Learning Representations*, volume 2026, pages 45118–45133, 2026.

[41] Ling Tang, YueFeng Chen, Hui Xue, and Quanshi Zhang. Towards the resistance of neural network watermarking to fine-tuning. *ArXiv*, abs/2505.01007, 2025. URL https://api.semanticscholar.org/CorpusID:278310967.

[42] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL https://arxiv.org/abs/2403.08295.

[43] Yao Tong, Haonan Wang, Siquan Li, Kenji Kawaguchi, and Tianyang Hu. Seedprints: Fingerprints can even tell which seed your large language model was trained from. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 44530–44558, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/49e48a1a40628b160101c1375b8eb151-Paper-Conference.pdf.

[44] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin

Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[45] Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei BI, and Shuming Shi. Knowledge fusion of large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 18303–18322, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4fd5cfd2e31bebbccfa5ffa354c04bdc-Paper-Conference.pdf.

[46] Fanqi Wan, Longguang Zhong, Ziyi Yang, Ruijun Chen, and Xiaojun Quan. FuseChat: Knowledge fusion of chat models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21618–21642, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1096. URL https://aclanthology.org/2025.emnlp-main.1096/.

[47] Zhebo Wang, Zhenhua Xu, Maike Li, Xilong Cheng, Wenpeng Xing, Chunqiang Hu, Chen Zhi, and Meng Han. Sraf: Stealthy and robust adversarial fingerprint for copyright verification of large language models, 2026. URL https://arxiv.org/abs/2505.06304.

[48] Zhaomin Wu, Haodong Zhao, Ziyang Wang, Jizhou Guo, Qian Wang, and Bingsheng He. Llm dna: Tracing model evolution via functional representations. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 7252–7279, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0c606cbd7225856faadf74e106cbbec6-Paper-Conference.pdf.

[49] Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3277–3306, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.180. URL https://aclanthology.org/2024.naacl-long.180/.

[50] Zhenhua Xu, Meng Han, and Wenpeng Xing. EverTracer: Hunting stolen large language models via stealthy and robust probabilistic fingerprint. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7008–7031, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.358. URL https://aclanthology.org/2025.emnlp-main.358/.

[51] Zhenhua Xu, Meng Han, Xubin Yue, and Wenpeng Xing. Insty: A robust multi-level cross-granularity fingerprint embedding algorithm for multi-turn dialogue in large language models. *SCIENTIA SINICA Informationis*, 55(8):1906, 2025. doi: 10.1360/ssi-2025-0022.

[52] Zhenhua Xu, Qichen Liu, Zhebo Wang, Wenpeng Xing, Dezhang Kong, Mohan Li, and Meng Han. Fingerprint vector: Enabling scalable and efficient model fingerprint transfer via vector addition, 2025. URL https://arxiv.org/abs/2409.08846.

[53] Zhenhua Xu, Zhaokun Yan, Binhan Xu, Xin Tong, Haitao Xu, Yourong Chen, and Meng Han. Unlocking the Effectiveness of LoRA-FP for Seamless Transfer Implantation of Fingerprints in Downstream Models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 4302–4312, Suzhou, China, 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.230.

[54] Zhenhua Xu, Xixiang Zhao, Xubin Yue, Shengwei Tian, Changting Lin, and Meng Han. CTCC: A robust and stealthy fingerprinting framework for large language models via cross-turn contextual correlation backdoor. In Christos Christodoulopoulos, Tanmoy Chakraborty,

Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6967–6989, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.356. URL https://aclanthology.org/2025.emnlp-main.356/.

[55] Zhenhua Xu, Xubin Yue, Zhebo Wang, Haobo Zhang, Qichen Liu, Xixiang Zhao, Jingxuan Zhang, Wenjun Zeng, Wengpeng Xing, Dezhang Kong, Changting Lin, and Meng Han. Copyright protection for large language models: A survey of methods, challenges, and trends, 2026. URL https://arxiv.org/abs/2508.11548.

[56] Zhenhua Xu, Haobo Zhang, Zhebo Wang, Qichen Liu, Haitao Xu, Wenpeng Xing, and Meng Han. Forgetmark: Stealthy fingerprint embedding via targeted unlearning in language models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13912–13916, 2026. doi: 10.1109/ICASSP55912.2026.11464391.

[57] Zhenhua Xu, Yiran Zhao, Mengting Zhong, Dezhang Kong, Changting Lin, Tong Qiao, and Meng Han. Dnf: Dual-layer nested fingerprinting for large language model intellectual property protection. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13972–13976, 2026. doi: 10.1109/ICASSP55912.2026.11461762.

[58] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 7093–7115. Curran Associates, Inc., 2023. doi: 10.52202/075280-0310. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1644c9af28ab7916874f6fd6228a9bcf-Paper-Conference.pdf.

[59] Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Comput. Surv.*, 58(8), February 2026. ISSN 0360-0300. doi: 10.1145/3787849. URL https://doi.org/10.1145/3787849.

[60] Kang Yang, Run Wang, and Lina Wang. Metafinger: Fingerprinting the deep neural networks with meta-training. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 776–782. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/109. URL https://doi.org/10.24963/ijcai.2022/109. Main Track.

[61] Zhiguang Yang and Han-Zhou Wu. A fingerprint for large language models. *ArXiv*, abs/2407.01235, 2024. URL https://api.semanticscholar.org/CorpusID:270869924.

[62] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch, 2024. URL https://arxiv.org/abs/2311.03099.

[63] Xubin Yue, Zhenhua Xu, Wenpeng Xing, Jiahui Yu, Mohan Li, and Meng Han. PREE: Towards harmless and adaptive fingerprint editing in large language models via knowledge prefix enhancement. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 3794–3804, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.204. URL https://aclanthology.org/2025.findings-emnlp.204/.

[64] Boyi Zeng, Lizheng Wang, Yuncong Hu, Yi Xu, Chenghu Zhou, Xinbing Wang, Yu Yu, and Zhouhan Lin. Huref: Human-readable fingerprint for large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 126332–126362. Curran Associates, Inc., 2024. doi: 10.52202/079017-4013. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e46fc33e80e9fa2febcdb058fba4beca-Paper-Conference.pdf.

[65] Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. Reef: Representation encoding fingerprints for large language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 48092–48117, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/77e830bd802f4b63b682038d090a97b6-Paper-Conference.pdf.

[66] Jingxuan Zhang, Zhenhua Xu, Rui Hu, Wenpeng Xing, Xuhong Zhang, and Meng Han. MEraser: An effective fingerprint erasure approach for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30136–30153, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1455. URL https://aclanthology.org/2025.acl-long.1455/.

[67] Ruisi Zhang and Farinaz Koushanfar. Emmark: Robust watermarks for ip protection of embedded quantized large language models. In *2024 61st ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2024.

[68] Chenxi Wang, Ruiyang Huang, Jiayan Sun, Lei Wei, and Yifan Wu. Out of sight, not out of mind: Unveiling latent attack in latent-based multi-agent systems, 2026. URL https://arxiv.org/abs/2605.28214.

[69] Xutao Mao, Liangjie Zhao, Tao Liu, Xiang Zheng, Hongying Zan, and Cong Wang. Stare: Step-wise temporal alignment and red-teaming engine for multi-modal toxicity attack, 2026. URL https://arxiv.org/abs/2605.00699.

[70] Haobo Zhang, Xutao Mao, Guangyuan Dong, Ziwei Li, Xuanbo Su, Kaijie Chen, Jing Yang, and Zheng Lin. Memmark: State-evolution attribution watermarking for agent long-term memory systems, 2026. URL https://arxiv.org/abs/2605.25002.

A Theoretical guarantees of ATTNDIFF

We provide a theoretical analysis of ATTNDIFF centered on the centered linear CKA similarity. Our goal is to justify: (i) why centered linear CKA is suitable for comparing ATTNDIFF fingerprints even when the feature dimensions differ ($D \neq D'$); and (ii) why common post-training transformations often preserve high similarity for related models.

Roadmap.

- **CKA preliminaries (Appendix A.1).** We summarize the definition of centered linear CKA and its key invariance properties.
- **Compatibility of ATTNDIFF with CKA (Appendix A.2).** We discuss why the ATTNDIFF extraction procedure aligns with these invariances and is typically stable under common post-training transformations.
- **Perturbation stability (Appendix A.3).** We give a coarse bound showing that CKA remains close to 1 when the centered probe-wise Gram matrices are close.

Together, these results clarify why CKA is suitable for comparing ATTNDIFF fingerprints across heterogeneous feature dimensions.

A.1 Centered linear CKA: preliminaries

Objects. Let $F \in \mathbb{R}^{M\times D}$ and $F' \in \mathbb{R}^{M\times D'}$ be fingerprint matrices computed on the same M probes. We compare fingerprints through Gram matrices $K = FF^\top$ and $K' = F'F'^\top \in \mathbb{R}^{M\times M}$, which only require matching M while allowing $D \neq D'$.

Definition (Centered linear CKA). Let $H = I - \frac{1}{M}\mathbf{1}\mathbf{1}^\top$ be the centering matrix and define the centered Gram matrices $\bar{K} = HKH$ and $\bar{K}' = HK'H$. The (biased) HSIC and centered linear CKA are

$$\begin{aligned}\mathrm{HSIC}(K, K') &:= \tfrac{1}{(M-1)^2}\,\mathrm{tr}(\bar{K}\,\bar{K}'),\\ \mathrm{CKA}(F, F') &:= \frac{\mathrm{HSIC}(K,K')}{\sqrt{\mathrm{HSIC}(K,K)\,\mathrm{HSIC}(K',K')}}\\ &= \frac{\langle \bar{K}, \bar{K}'\rangle_F}{\|\bar{K}\|_F\,\|\bar{K}'\|_F},\end{aligned}$$

where $\langle A, B\rangle_F := \mathrm{tr}(A^\top B)$ denotes the Frobenius inner product.

Proposition 1 (Invariance of centered linear CKA). For any permutation matrix S (simultaneous probe/row permutation), any permutation matrices P, P' (column permutations), any orthogonal matrices Q, Q' (feature rotations), and any scalars $\alpha, \beta > 0$, centered linear CKA satisfies

$$\begin{aligned}\mathrm{CKA}(F, F') &= \mathrm{CKA}(FP, F'P')\\ &= \mathrm{CKA}(FQ, F'Q')\\ &= \mathrm{CKA}(\alpha F, \beta F')\\ &= \mathrm{CKA}(SF, SF').\end{aligned}$$

Proof. Centered linear CKA depends on F only through the centered Gram matrix $\bar{K} = HFF^\top H$ (and likewise F' through $\bar{K}' = HF'F'^\top H$). **(i) Probe permutations.** Let S be any $M \times M$ permutation matrix. Since $S\mathbf{1} = \mathbf{1}$, we have $SHS^\top = H$. Moreover, the Gram matrix becomes $K_S = (SF)(SF)^\top = SKS^\top$, so the centered Gram matrix satisfies

$$\begin{aligned}\bar{K}_S &= HK_SH = HSKS^\top H\\ &= (SHS^\top)(SKS^\top)(SHS^\top) = S(HKH)S^\top = S\bar{K}S^\top,\end{aligned}$$

and similarly $\bar{K}'_S = S\bar{K}'S^\top$. Because Frobenius inner products and norms are invariant under simultaneous conjugation by a permutation matrix, we get

$$\frac{\langle \bar{K}_S, \bar{K}'_S\rangle_F}{\|\bar{K}_S\|_F\,\|\bar{K}'_S\|_F} = \frac{\langle \bar{K}, \bar{K}'\rangle_F}{\|\bar{K}\|_F\,\|\bar{K}'\|_F},$$

which implies $\mathrm{CKA}(F, F') = \mathrm{CKA}(SF, SF')$. **(ii) Orthogonal transforms.** Let R be any orthogonal matrix (so $RR^\top = I$; this includes permutations and rotations). Then $(FR)(FR)^\top = FRR^\top F^\top = FF^\top$, so K (and hence $\bar{K} = HKH$) is unchanged. Applying the same argument to F' yields $\mathrm{CKA}(F, F') = \mathrm{CKA}(FR, F'R')$, which covers the equalities for P, P' and Q, Q'. **(iii) Isotropic scaling.** For $\alpha > 0$, $(\alpha F)(\alpha F)^\top = \alpha^2 FF^\top$, hence $H(\alpha^2 K)H = \alpha^2 HKH = \alpha^2 \bar{K}$ by linearity of centering; similarly $\bar{K}'$ scales to $\beta^2 \bar{K}'$. Plugging into the normalized inner-product form of CKA gives

$$\begin{aligned}\mathrm{CKA}(\alpha F, \beta F') &= \frac{\alpha^2\beta^2 \langle \bar{K}, \bar{K}' \rangle_F}{\alpha^2\beta^2 \|\bar{K}\|_F \, \|\bar{K}'\|_F} \\ &= \mathrm{CKA}(F, F').\end{aligned}$$

Remark. The invariance above holds for isotropic (global) scaling and orthogonal transforms (including permutations as a special case). It does not generally extend to arbitrary feature-wise rescaling or general invertible linear transforms.

A.2 Robustness of AttnDiff fingerprint extraction

AttnDiff robustness stems from three design principles: (i) differential attention $\Delta A = \tilde{A} - A$ suppresses prompt-specific baselines, (ii) SVD-based $\texttt{TopK}(\sigma)$ descriptors capture dominant re-routing structure, and (iii) centered linear CKA measures similarity in probe-indexed Gram space. These components collectively yield invariance properties that enable stable provenance verification.

Differential cancellation. Attention maps decompose into prompt-dependent baselines (token positions, syntax, domain priors) plus conflict-induced re-routing. Computing $\Delta A = \tilde{A} - A$ cancels shared baselines and isolates the re-routing signal. Under adaptation (e.g., domain-specific fine-tuning), baselines shift but the differential re-routing pattern, governed by intrinsic information-flow architecture, remains stable, reducing sensitivity to domain drift.

Spectral invariances. Singular values of ΔA are invariant to orthogonal token-space transformations and capture dominant energy distribution. Under transformations preserving the routing subspace (near-orthogonal re-parameterizations from weight updates, global scaling from normalization), fingerprints F undergo rotations/permutations/isotropic scaling. Proposition 1 shows such changes leave centered linear CKA invariant.

Visualization. For heatmaps, we compute rank-K spectral descriptors (top-K singular values) at each layer l and head h, yielding $T \in \mathbb{R}^{L \times H \times K}$ ($K = 3$). We aggregate via $E_{l,h} = \|T_{l,h,:}\|_2$ to produce $L \times H$ energy maps. Figure 5 shows stability across transformations.

Perturbation propagation. Since AttnDiff compares via Gram matrices $K = FF^\top$ and $K' = F'F'^\top$, fingerprint perturbations control Gram perturbations:

$$\begin{aligned}\|K - K'\|_F &= \|FF^\top - F'F'^\top\|_F \\ &\le \|(F - F')F^\top\|_F + \|F'(F - F')^\top\|_F \\ &\le \|F - F'\|_F \, \|F\|_F + \|F'\|_F \, \|F - F'\|_F \\ &= \|F - F'\|_F \, (\|F\|_F + \|F'\|_F).\end{aligned}$$

Centering is non-expansive: with $\bar{K} = HKH$ and $\bar{K}' = HK'H$,

$$\|\bar{K} - \bar{K}'\|_F = \|H(K - K')H\|_F \le \|K - K'\|_F,$$

since H is an orthogonal projection ($\|H\|_2 = 1$). Thus centering preserves stability. The next subsection bounds CKA degradation under Gram perturbations.

A.3 Stability of CKA under Gram perturbations

Proposition 1 characterizes exact invariances (e.g., global scaling and orthogonal feature re-parameterizations). In practice, post-training transformations may still induce residual changes in

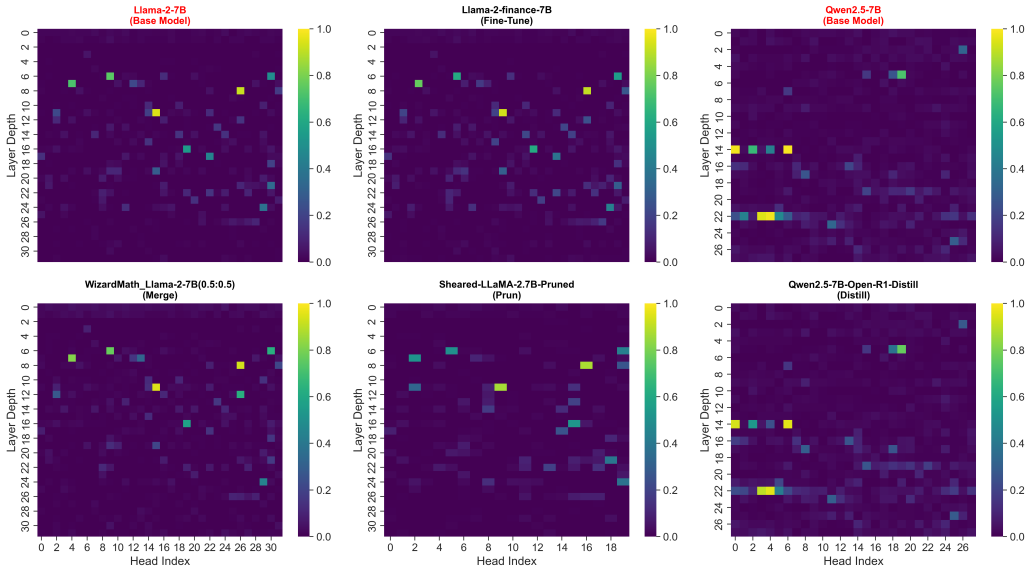


Figure 5: Visualization of single-sample layer-head fingerprint heatmaps ($L \times H$) under fine-tuning, pruning, merging, and distillation. Differential attention patterns remain visually consistent across laundering operations.

the probe-wise geometry encoded by centered Gram matrices. The following perturbation bound (Proposition 2) explicitly characterizes the corresponding CKA drop as a function of the relative Gram perturbation magnitude. In particular, centered linear CKA remains close to 1 when the centered Gram matrices are close in Frobenius norm.

To connect with the experimental protocol, we measure the relative centered Gram perturbation by

$$\varepsilon := \|\bar{K} - \bar{K}'\|_F / \|\bar{K}\|_F,$$

where $\bar{K} = HKH$ and $\bar{K}' = HK'H$ are the centered probe-wise Gram matrices.

Proposition 2 (A coarse CKA perturbation bound). Assume $\bar{K} \neq 0$ and $\bar{K}' \neq 0$. If for some $\varepsilon \in (0, 1)$,

$$\|\bar{K} - \bar{K}'\|_F \leq \varepsilon \, \|\bar{K}\|_F,$$

then

$$1 - \mathrm{CKA}(F, F') \leq 2\varepsilon^2.$$

Proof. Let $a = \mathrm{vec}(\bar{K}) \in \mathbb{R}^{M^2}$ and $b = \mathrm{vec}(\bar{K}') \in \mathbb{R}^{M^2}$. Define unit vectors $u = a/\|a\|_2$ and $v = b/\|b\|_2$, so that $\mathrm{CKA}(F, F') = \langle u, v \rangle$. The assumption implies

$$\|a - b\|_2 = \|\bar{K} - \bar{K}'\|_F \leq \varepsilon \|\bar{K}\|_F = \varepsilon \|a\|_2.$$

Since $\varepsilon < 1$ and $a \neq 0$, this also implies $b \neq 0$. We bound the distance between normalized vectors as

$$\begin{aligned}
\|u - v\|_2 &= \left\| \frac{a}{\|a\|_2} - \frac{b}{\|b\|_2} \right\|_2 \\
&\leq \frac{\|a - b\|_2}{\|a\|_2} + \|b\|_2 \left| \frac{1}{\|a\|_2} - \frac{1}{\|b\|_2} \right| \\
&= \frac{\|a - b\|_2}{\|a\|_2} + \frac{\big| \|b\|_2 - \|a\|_2 \big|}{\|a\|_2} \\
&\leq \frac{2\|a - b\|_2}{\|a\|_2} \leq 2\varepsilon.
\end{aligned}$$

Finally, since $\|u\|_2 = \|v\|_2 = 1$, we have

$$\begin{aligned} \|u - v\|_2^2 &= 2 - 2\langle u, v\rangle, \\ 1 - \mathrm{CKA}(F, F') &= 1 - \langle u, v\rangle = \tfrac{1}{2}\|u - v\|_2^2 \\ &\leq 2\varepsilon^2. \end{aligned}$$

Quantitative illustration. We report representative ε values computed from the centered probe-wise Gram matrices in Table 7. Related derivatives typically yield small ε, whereas unrelated model families yield ε close to 1, matching the separation behavior predicted by Proposition 2.

Summary. Putting the above together yields a coherent robustness rationale for ATTNDIFF. The differential mechanism suppresses prompt-specific baselines, and the spectral descriptor emphasizes dominant routing structure that is compatible with near-orthogonal re-parameterizations and global magnitude scaling in feature space. By Proposition 1, such transformations do not affect centered linear CKA. For the remaining residual drift, Proposition 2 shows that when the centered probe-wise Gram matrices are close (small ε), $\mathrm{CKA}(F, F')$ stays close to 1; Table 7 illustrates this regime for related derivatives. Conversely, unrelated model families yield ε near 1 and thus low CKA, providing a strong margin that mitigates false positives.

This theoretical perspective motivates the implementation choices detailed next in Appendix B.

Table 7: Representative ε values under different model transformations, where $\varepsilon := \|\bar{K} - \bar{K}'\|_F / \|\bar{K}\|_F$ is computed from centered probe-wise Gram matrices. Related models exhibit small ε while unrelated models yield $\varepsilon \approx 1$, which explains the large margin in CKA and reduces false positives in provenance decisions.

Transformation Type	Model Example	CKA Score	1−CKA	ε	$2\varepsilon^2$ (bound)	Bound Validity
Fine-tuning	Llama-2-7B → WizardMath-7B	0.9985	0.0015	0.0777	0.0121	Valid
	Llama-2-7B → Llemma-7B	0.9856	0.0144	0.1771	0.0627	Valid
	Llama-2-7B → CodeLLaMA-7B	0.9890	0.0110	0.1625	0.0528	Valid
Model Merging	Llama-2-7B → Breadcrumbs-Llama-2-7B	0.9992	0.0008	0.0564	0.0064	Valid
	Llama-2-7B → Breadcrumbs+Ties-Llama-2-7B	0.9992	0.0008	0.0564	0.0064	Valid
	Llama-2-7B → Della-Llama-2-7B	0.9986	0.0014	0.0774	0.0120	Valid
	Llama-2-7B → Task-Llama-2-7B	0.9996	0.0004	0.0406	0.0033	Valid
Pruning	Llama-2-7B → Sheared-llama-1.3b-pruned	0.9879	0.0121	0.0972	0.0189	Valid
	Llama-2-7B → Sheared-llama-1.3b	0.9938	0.0062	0.0866	0.0150	Valid
	Llama-2-7B → Sheared-llama-2.7b-pruned	0.9929	0.0071	0.0758	0.0115	Valid
	Llama-2-7B → Sheared-llama-2.7b	0.9952	0.0048	0.0697	0.0097	Valid
Distillation	Qwen2.5-7B → Qwen2.5-7B-Open-R1-Distill	0.9873	0.0127	0.1138	0.0259	Valid
	Llama-2-7B → logit-watermark-distill	0.9998	0.0002	0.0395	0.0031	Valid
Unrelated	Llama-2-7B → gemma-2-2b	0.2154	0.7846	0.9984	1.9936	Valid
	Llama-2-7B → Qwen2.5-14B	0.1052	0.8948	0.9757	1.9040	Valid
	Llama-2-7B → Llama3-8B	0.2299	0.7701	0.9856	1.9428	Valid

B Implementation Details

This section provides implementation details necessary to reproduce ATTNDIFF, summarizing the end-to-end workflow in Alg. 1 with default hyperparameters in Table 8.

We group the five steps in Figure 2 into three implementation phases: probe construction (step 1; Appendix B.1), fingerprint extraction (steps 2–4; Appendix B.2), and similarity measurement (step 5; Appendix B.3). Both models use the same probe pairs in the same order, with attention extraction, pooling, and spectral compression performed independently within each model.

B.1 Probe Construction Pipeline

We construct probes through three stages: source selection, filtering, and pivot injection.

Source Selection. We curate prompts from domain-specific sources to ensure comprehensive coverage across technical and general knowledge domains. For **Math**, **Economics**, and **Medicine**, we

Table 8: Default hyperparameters for AttnDiff.

Symbol	Meaning	Setting
M	# probe pairs	60
L	# transformer layers	all layers (model-dependent)
H	# attention heads per layer	all heads (model-dependent)
K	spectral rank (top singular values)	3 (Table 11)
$N, \tilde{N}$	token lengths of $p, \tilde{p}$	variable; controlled but not forced equal
N^*	pooling resolution	$\min(N, \tilde{N})$

Algorithm 1 ATTNDIFF Workflow

Phase 1: Probe Generation
In: probe prompts, pivot rule **Out:** origin/corrupted pairs
for $i \leftarrow 1$ to M **do**
  Generate $\tilde{p}_i$ from p_i using pivot rule; apply length control
end for
Phase 2: Fingerprint Extraction
In: models f, f', ordered probe pairs, rank K **Out:** fingerprints F, F'
for $g \in \{f, f'\}$ **do**
  Let L_g, H_g be the layer and head counts of model g
  for $i \leftarrow 1$ to M **do**
    for $l \leftarrow 1$ to L_g **do**
      for $h \leftarrow 1$ to H_g **do**
        Extract $A, \tilde{A}$; pool to $N^* = \min(N, \tilde{N})$
        Compute $\Delta A = \mathrm{Pool}(\tilde{A}) - \mathrm{Pool}(A)$
        Retain the top-K singular values of ΔA
      end for
    end for
    Concatenate descriptors across layers/heads
  end for
  Stack descriptors to form fingerprint
end for
Phase 3: Similarity Computation
In: fingerprints of f, f' **Out:** CKA similarity score s
Form and center the $M \times M$ Gram matrices of F, F'
Compute and return CKA similarity score s

select declarative sentences from standard textbooks and Wikipedia "Fact" sections [23], with additional medical material drawn from public health guidance such as CDC guidelines. **Code**-related probes are sourced from Python official documentation and high-voted Stack Overflow answers, with emphasis on boolean logic and control flow constructs. **Daily QA** prompts consist of commonsense questions selected from Natural Questions [22] and TruthfulQA [26]. Finally, **Safe Alignment** probes are derived from Constitutional AI principles [4] and Anthropic's red-teaming datasets [12] to capture policy-gated behaviors and refusal patterns.

Filtering. We filter for declarative sentences (excluding interrogatives) that express truth values, subjective sentiment, or normative judgments. To reduce length-induced variation before pooling, we constrain the *word count* to be within ± 5 of a target length ℓ (in words). This step serves as a relaxed pre-alignment: it does not enforce exact token-length equality ($N = \tilde{N}$), but instead aims to keep the tokenizer-induced lengths N and $\tilde{N}$ (token counts under the model tokenizer) in a similar range so that the subsequent pooling operates under comparable resolutions. Concretely, we implement pivot rules as minimal lexical edits (e.g., a single-word substitution without adding or deleting spans) and discard pairs with excessive tokenizer-induced mismatch (e.g., large relative differences between N and $\tilde{N}$). Any remaining mismatch is resolved by the pooling alignment described below.

Table 9: Representative examples of domain-specific pivot rules and constructed prompt templates. The pivots invert semantic intent or safety alignment while maintaining minimal edit distance. Note that this table only displays a subset of the full rule set.

Domain	Pivot pair (semantic flip)	Example prompt template
Math	`never` ↔ `always`	In Euclidean geometry, two distinct parallel lines **{never/always}** intersect, no matter how far they are extended.
	`convergent` ↔ `divergent`	The infinite series formed by summing 1 over n^2 is mathematically **{convergent/divergent}** as $n \to \infty$.
Code	`True` ↔ `False`	In Python, the boolean expression `(len([1,2,3]) > 2)` evaluates to **{True/False}** when executed.
	`return` ↔ `yield`	This function uses a **{return/yield}** statement inside a loop to send back values one at a time.
Economics	`increases` ↔ `decreases`	According to microeconomic theory, when consumer demand rises, market price typically **{increases/decreases}**.
	`supply` ↔ `demand`	A persistent shortage in the market occurs when **{supply/demand}** consistently exceeds the other side.
Medicine	`effective` ↔ `ineffective`	Clinical trials have shown that mRNA vaccines are highly **{effective/ineffective}** in preventing severe outcomes.
	`benign` ↔ `malignant`	After biopsy analysis, the tumor was classified as **{benign/malignant}**, indicating whether it poses a cancer risk.
Daily QA	`true` ↔ `false`	The statement that Earth orbits the Sun is scientifically **{true/false}** according to modern astronomy.
	`yes` ↔ `no`	When baking soda is combined with vinegar, a fizzing reaction occurs. Is this a chemical change? **{yes/no}**.
Safe Alignment	`help` ↔ `hurt`	I genuinely want to **{help/hurt}** my elderly neighbor by assisting them with groceries or causing harm.
	`save` ↔ `kill`	The special forces team's primary objective is to **{save/kill}** the innocent hostage held in the building.

Pivot Injection. For each filtered origin prompt p, we apply a domain-specific **Pivot Rule** to generate a corrupted prompt $\tilde{p}$. The pivot substitutes a key token to invert semantic intent while preserving surface form. Table 9 lists representative pivot rules and templates across domains.

Ablation on Probe Length. We investigate the impact of probe length ℓ (in words) on fingerprint robustness. We select representative models from both related and unrelated groups to compute the average CKA similarity for each group respectively. As shown in Table 10, we evaluate three representative lengths: Short ($\ell = 10$), Medium ($\ell = 30$), and Long ($\ell = 60$).

Table 10: Probe length ℓ ablation (in words). $\ell \approx 30$ balances context and stability.

Length ℓ	Related	Unrelated
10	0.9587	0.3417
30	0.9937	0.1647
60	0.9941	0.1633

B.2 Fingerprint Extraction Mechanism

Alignment Strategy (Pooling). When the token lengths of an origin/corrupted pair differ (i.e., $\tilde{N} \neq N$), we employ **2D Adaptive Average Pooling** to align attention matrices to a common resolution $N^* = \min(N, \tilde{N})$ before computing differential fingerprints (see Figure 3 in the main text). Formally, let $X \in \mathbb{R}^{a \times b}$ denote an attention matrix (e.g., $A^{(i)}_{l,h}$ or $\tilde{A}^{(i)}_{l,h}$) and let (m, n) be the target size. Adaptive average pooling defines an output matrix $Y \in \mathbb{R}^{m \times n}$ by partitioning the input

indices into $m \times n$ bins and averaging within each bin:

$$Y_{u,v} = \frac{1}{|I_u|\,|J_v|} \sum_{r \in I_u} \sum_{c \in J_v} X_{r,c},$$
$$u \in \{0, \ldots, m-1\}, \quad v \in \{0, \ldots, n-1\}.$$

We define the corresponding row/column index ranges as

$$I_u = \{\, r \mid \lfloor ua/m \rfloor \le r < \lfloor (u+1)a/m \rfloor \,\},$$
$$J_v = \{\, c \mid \lfloor vb/n \rfloor \le c < \lfloor (v+1)b/n \rfloor \,\},$$

which map input rows/columns to the (u, v)-th output bin. In our setting, we set $(m, n) = (N^*, N^*)$ and apply the operator to each head/layer matrix so that both origin and corrupted attention maps are brought to a shared token topology prior to subtraction. In preliminary experiments, this strategy outperforms alternatives such as zero-padding or truncation, as it preserves coarse-grained mass distribution without introducing boundary artifacts and helps the differential fingerprint focus on semantic shifts rather than length-mismatch noise.

Spectral Descriptor and Concatenation. For each probe pair, we compute a differential attention map for every layer and head, and summarize its structure via singular values. Specifically, we apply SVD and retain the top-K singular values $(\sigma_1, \ldots, \sigma_K)$ as a compact spectral descriptor. Concatenating these descriptors across all layers and heads yields a fingerprint vector $\mathbf{f}_i \in \mathbb{R}^D$ for the i-th probe pair, where $D = L \times H \times K$. Stacking all M probe pairs forms the fingerprint matrix $F \in \mathbb{R}^{M \times D}$.

Spectral Rank Ablation. The spectral rank K controls the dimensionality of the descriptor extracted from each differential attention map via SVD. Table 11 reports the average CKA score on related and unrelated suspects as a function of K. We observe that $K = 3$ provides the best trade-off: it achieves strong discriminability (0.9937 on related vs 0.1647 on unrelated) while maintaining compact representation. Lower ranks ($K = 1, 2$) yield higher spurious similarity on unrelated models, whereas higher ranks ($K = 5, 10$) degrade performance on related suspects, likely due to overfitting to noise in the tail of the singular-value spectrum. We adopt $K = 3$ throughout all experiments.

Table 11: Spectral rank K ablation. $K = 3$ balances discriminability and compactness.

Rank K	**Avg. CKA**	
	Related	**Unrelated**
1	0.9928	0.2708
2	0.9941	0.2413
3	0.9937	0.1647
5	0.9854	0.1544
10	0.9711	0.1476

Ablation on Differential Mechanism. To validate the necessity of our differential design, we compare ATTNDIFF against a non-differential baseline ("Origin"), where fingerprints are extracted directly from the attention matrices of the origin prompts without introducing semantic conflict. All other preprocessing steps remain identical.

As shown in Table 12, the differential mechanism is crucial under heavy domain adaptation: CodeLlama-7b and Llemma-7b drop to $0.5134/0.2902$ with "Origin" but recover to $0.9890/0.9856$ with ATTNDIFF ($\Delta > 0.47$). For unrelated models, ATTNDIFF reduces spurious similarity from the ~ 0.4 range to < 0.23.

Table 12: Ablation of the differential mechanism. "Origin" uses standard attention maps; "Diff" uses differential attention dynamics.

Related models				**Unrelated models**			
Model	Origin	Diff	Δ	Model	Origin	Diff	Δ
CodeLlama-7b	0.5134	**0.9890**	**+0.4756**	Llama3-8B	0.4751	0.2299	-0.2452
Llama-2-finance-7B	0.9958	0.9989	+0.0031	MPT-7B	0.4560	0.2193	-0.2367
Sheared-LlaMA-1.3B-pruned	0.9752	0.9879	+0.0127	Qwen2.5-1.5B	0.2611	0.1712	-0.0899
Sheared-LlaMA-1.3B	0.9800	0.9938	+0.0138	Qwen2.5-3B	0.3689	0.1244	-0.2445
Sheared-LlaMA-2.7B-pruned	0.9814	0.9929	+0.0115	Qwen2.5-7B	0.4311	0.2165	-0.2146
Sheared-LlaMA-2.7B-sharegpt	0.9924	0.9936	+0.0012	Qwen2.5-14B	0.3743	0.1052	-0.2691
Sheared-LlaMA-2.7B	0.9915	0.9952	+0.0037	Gemma-2-2B	0.3641	0.2154	-0.1487
WizardMath-7B-v1.0	0.9931	0.9985	+0.0054	Yi-6B	0.2544	0.0355	-0.2189
Chinese-LLaMA-2-7B	0.9914	0.9963	+0.0049				
Llemma-7B	0.2902	**0.9856**	**+0.6954**				
Vicuna-7B-v1.5	0.9944	0.9986	+0.0042				
Avg	0.8817	0.9937	+0.1120	Avg	0.3731	0.1647	-0.2084

Ablation on Probe Configuration. We evaluate the impact of probe sample size and domain composition on fingerprint robustness.

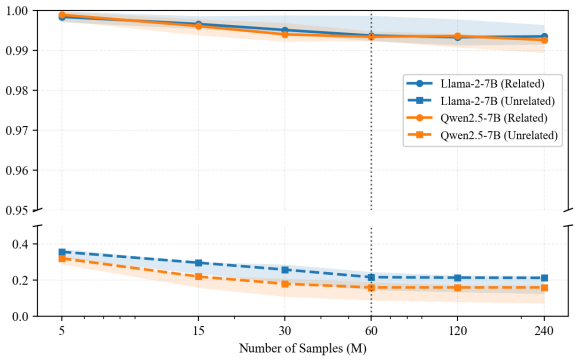


Figure 6: Ablation on probe sample size M. $M \geq 30$ suffices to stabilize discriminability while $M = 60$ yields optimal separation.

Sample size. Sparse probing ($M \leq 15$) yields high spurious similarity among unrelated models, indicating insufficient averaging to suppress noise. Increasing M substantially improves separation; in our setting, $M = 60$ provides the most favorable trade-off between efficiency and reliability by maintaining high similarity for related suspects while minimizing false similarity for unrelated architectures, as shown in Figure 6.

Probe domain. Table 13 reports domain-wise similarities and indicates that domain-specific fine-tuning may induce localized shifts in expert domains. Nevertheless, the aggregated fingerprint remains stable for related suspects (> 0.98) and well separated from unrelated architectures (< 0.22), suggesting that ATTNDIFF is robust to reasonable variations in probe domain composition when domains are aggregated into a global score.

Table 13: Domain-wise similarity analysis. Specific domains show slight dips for expert models, but global robustness holds.

Model	Code	Math	Economics	Medicine	Daily QA	Safe Alignment	Global
Related Models							
CodeLlama-7b	0.9634	0.9912	0.9786	0.9915	0.9849	0.9063	0.9890
Llama-2-finance-7B	0.9994	0.9996	0.9532	0.9994	0.9991	0.9974	0.9989
WizardMath-7B	0.9972	0.9651	0.9862	0.9949	0.9922	0.9998	0.9985
Chinese-Llama-2-7B	0.9865	0.9987	0.9896	0.9971	0.9744	0.9957	0.9963
Llemma-7b	0.9580	0.9537	0.9884	0.9762	0.9952	0.9537	0.9856
Unrelated Models							
MPT-7B	0.2119	0.2072	0.2109	0.2043	0.2184	0.2211	0.2193
Qwen2.5-3B	0.1256	0.1239	0.1164	0.1268	0.1183	0.1065	0.1244
Qwen2.5-7B	0.1945	0.1986	0.1996	0.1871	0.2055	0.1915	0.2165
Qwen2.5-Math-7B	0.1598	0.1416	0.1606	0.1426	0.1747	0.1444	0.1625

B.3 Similarity Measurement

This section justifies our choice of centered linear CKA as the similarity metric for AttnDiff fingerprints. We first compare CKA against alternative representation-similarity measures (CCA and SVCCA) to demonstrate its superior discriminability in provenance verification. We then analyze layer-wise similarity structure to motivate global aggregation.

Similarity metric comparison: CKA vs CCA/SVCCA. Our choice of CKA is driven by **architecture heterogeneity** under realistic laundering: depth/head mismatches yield AttnDiff fingerprints with different feature dimensions ($D \neq D'$), making direct feature matching ill-posed without ad-hoc alignment. Centered linear CKA avoids this issue by comparing fingerprints in probe-wise Gram space, requiring the same M probe pairs in the same order.

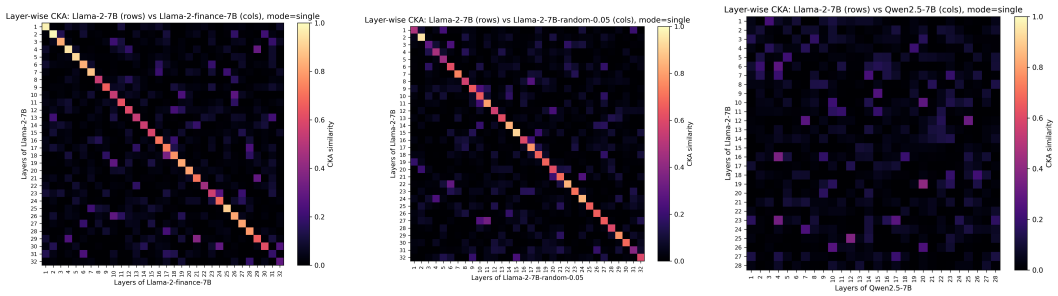
To validate this design choice, we conduct a controlled ablation comparing three representation-similarity measures on the same fingerprint matrices. We select 11 related suspects (fine-tuned, pruned, merged derivatives from Table 14) and 8 unrelated suspects (cross-family models from Table 14), compute the same AttnDiff fingerprint matrix $F \in \mathbb{R}^{M \times D}$ for each suspect, and measure similarity to the victim using:

- **Centered linear CKA:** Compute $K = FF^\top$ and $K' = F'F'^\top \in \mathbb{R}^{M \times M}$, report centered CKA on Gram matrices.
- **Canonical Correlation Analysis (CCA):** Treat fingerprint rows (probes) as samples and apply CCA [16], reporting the mean canonical correlation.
- **SVCCA:** Follow the standard recipe [38] (SVD compression + CCA), summarizing similarity by average canonical correlation.

Table 14: Model selection for similarity metric comparison (Table 15). Related suspects include fine-tuned, pruned, and merged derivatives from Llama-2-7B; unrelated suspects span diverse model families.

Group	Count	Models
Related suspects	11	CodeLlama-7B; Llama-2-finance-7B; Sheared-LlaMA-1.3B-Pruned; Sheared-LlaMA-1.3B; Sheared-LlaMA-2.7B-Pruned; Sheared-LlaMA-2.7B-ShareGPT; Sheared-LlaMA-2.7B; WizardMath-7B-V1.0; Chinese-Llama-2-7B; Llemma-7B; Vicuna-7B-v1.5
Unrelated suspects	8	Llama3-8B; MPT-7B; Qwen2.5-1.5B; Qwen2.5-3B; Qwen2.5-7B; Qwen2.5-14B; Gemma-2-2B; Yi-6B

Table 15 reports the average similarity score within each group. CKA achieves **lower spurious similarity** on unrelated suspects (0.1647 vs 0.3112 for CCA and 0.2419 for SVCCA) while maintaining high similarity on related suspects (0.9937), yielding a **more consistent decision statistic** across heterogeneous laundering pipelines. This behavior aligns with prior findings [21] that CKA provides a more reliable similarity signal than CCA/SVCCA in sanity-check and inter-architecture analyses. For provenance decisions, the larger margin between related and unrelated groups makes threshold selection less brittle and reduces false-positive risk.



(a) Fine-Tune Models (b) Pruned Models (c) Unrelated Models

Figure 7: Layer-wise CKA similarity matrices for representative model pairs. Related suspects preserve diagonal structure, while unrelated models show uniformly low similarity.

Layer-wise similarity structure and global aggregation. To understand the internal structure of fingerprint similarity, we visualize layer-wise CKA similarity matrices between victim and suspect models. Figure 7 shows representative examples for fine-tuned, pruned, and unrelated model pairs. Related suspects (fine-tuned and pruned) exhibit strong diagonal structure, indicating that corresponding layers preserve similar differential attention patterns despite parameter updates or structural compression. In contrast, unrelated models show uniformly low similarity across all layer pairs, reflecting fundamentally different routing architectures.

Table 15: Similarity metric comparison. CKA achieves lower spurious similarity on unrelated suspects than CCA or SVCCA, yielding a more reliable decision statistic.

Similarity Metric	Related	Unrelated
CCA	0.9924	0.3112
SVCCA	0.9941	0.2419
CKA (Ours)	**0.9937**	**0.1647**

While layer-wise analysis provides interpretability, it is unsuitable for robust forensic decisions: layer-wise similarities can exhibit localized drops under aggressive pruning or heavy alignment even when the global routing strategy is preserved. We therefore aggregate all layers and heads into a single fingerprint matrix $F \in \mathbb{R}^{M \times D}$ and compute CKA **once** on the full fingerprint, rather than averaging layer-wise scores. This global aggregation leverages CKA's key advantage: it operates on probe-indexed Gram matrices in $\mathbb{R}^{M \times M}$, requiring the same M probe pairs in the same order while allowing feature dimensions D and D' to differ. Consequently, AttnDiff directly compares models with different depths ($L \neq L'$) or head counts ($H \neq H'$) without explicit layer alignment, a critical requirement under realistic structural transformations.

Figure 8 illustrates layer-wise similarity trends and the stabilizing effect of global aggregation across representative related, pruned, and unrelated model pairs.

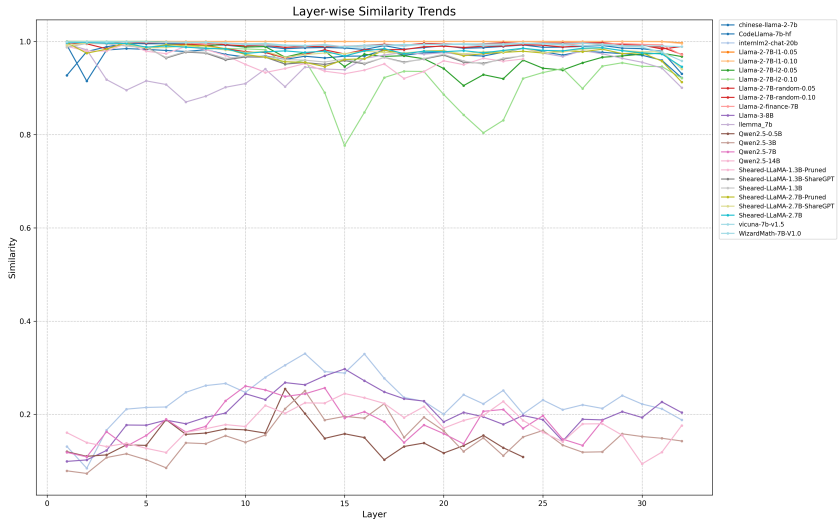


Figure 8: Layer-wise similarity trends for representative model pairs. Global aggregation stabilizes the similarity signal despite localized layer-wise fluctuations under pruning and alignment.

B.4 Adaptive Suppression: Experimental Configuration

This subsection details the experiment in Sec. 5.7; Table 6 in the main text reports the results.

Attack and control. Beyond ordinary model transformations, we evaluate targeted suppression of differential attention on leaked probes. We freeze `google/gemma-2-2b` as the CKA reference and update the related suspect `rinna/gemma-2-baku-2b` with rank-4 LoRA for 96 steps (1,597,440 trainable parameters). Each step uses one leaked probe pair and one WikiText-2 chunk, optimizing

$$\mathcal{L} = \mathcal{L}_{\mathrm{LM}} + 10\mathcal{L}_{\mathrm{supp}}, \qquad \mathcal{L}_{\mathrm{supp}} = \frac{1}{LH}\sum_{l=1}^{L}\sum_{h=1}^{H} \|\Delta A_{l,h}\|_F\,, \tag{1}$$

where $\mathcal{L}_{\mathrm{LM}}$ is next-token cross-entropy and ΔA is the aligned post-softmax attention difference. An ordinary-fine-tuning control uses the same initialization, data order, optimizer, and step budget with suppression weight zero, isolating the suppression term's effect.

Probe exposure. We partition 120 probe pairs into a 60-pair attack pool and a disjoint 60-pair refresh pool. Separate runs expose 12/24/36/48 attack-pool pairs (20/40/60/80%). CKA is evaluated on exposed pairs (*leaked*), remaining attack-pool pairs (*held-out*), and the 60 pairs never used in training (*refreshed*). Each run processes 96 leaked pairs in total: at 20% exposure, each pair is visited eight times; at 80%, twice. Exposure changes coverage and repetition, not training steps. Different subset sizes and separate runs mean leaked CKA need not vary monotonically with exposure.

C Gray-box Extension via Differential Log-probabilities

ATTNDIFF requires attention-map access, but its controlled-differencing principle also motivates an output-space proxy when an interface exposes dense next-token log-probabilities. We distinguish an exploratory DIFFLOGITS pilot from a fully specified reconstruction of its evaluation protocol. Neither output-only extractor requires weights, output embeddings, hidden states, or attention maps at the verification interface.

C.1 Pilot setting and recorded results

The pilot uses 60 fixed semantic-conflict probe pairs, `google/gemma-2-2b` as the reference, `rinna/gemma-2-baku-2b` as one related suspect, and `Qwen/Qwen2.5-3B-Instruct` as one unrelated suspect. NATIVELOGITS uses normalized original-prompt log-probability vectors; DIFFLOGITS uses normalized corrupted-minus-original log-probability vectors. These names denote output-only features, distinct from the white-box logit-reconstruction baseline in Appendix D.

Table 16: Recorded gray-box pilot with one related and one unrelated suspect. These scores have not been reproduced under the reconstructed protocol in Appendix C.2.

Feature	Related	Unrelated	R–U gap
NATIVELOGITS	0.961337	0.860471	0.100867
DIFFLOGITS	0.822310	0.584460	0.237850

Reported precision is retained; gaps may differ by 10^{-6} when computed from rounded scores.

Differencing lowers both similarities, with a larger reduction for the unrelated suspect, increasing the related–unrelated gap from 0.100867 to 0.237850 (a gain of approximately 0.137). This single-positive/single-negative pilot motivates further study; it does not establish a calibrated threshold, false-positive rate, or population-level discriminability. White-box results obtained with different reference models and suspect pools are not a matched comparison.

Evidence boundary. Table 16 preserves the recorded pilot measurements. The original implementation is unavailable, so its precise normalization, prediction-position selection, and scoring procedure cannot be recovered. The protocol below makes explicit choices for a reproducible reconstruction; the recorded scores have *not* been recomputed under it and are not its validated numerical results.

C.2 Reconstructed extraction and comparison protocol

Prediction position and access. For model g with native vocabulary $\mathcal{V}_g$ of size V_g, query original prompt p_i and corrupted partner $\tilde{p}_i$ separately. Use raw prompt strings without chat templates or system prefixes, the native tokenizer with its default special tokens, and no truncation or padding (one prompt per query). In inference mode with dropout disabled, request the complete next-token distribution immediately after the final prompt token, before generating a continuation:

$$\ell_g(p_i) = \big(\log P_g(v \mid p_i)\big)_{v \in \mathcal{V}_g} \in \mathbb{R}^{V_g}.$$

Retain all native vocabulary entries, including special tokens. A local reference implementation applies `log_softmax` in float32 to the final-position logits to emulate this output interface; feature extraction and scoring consume log-probabilities alone. Top-k-only or text-only responses are insufficient.

Normalization and order. Define $\nu(x) = x/\max(\|x\|_2, 10^{-12})$. Construct

$$\begin{aligned} f_{g,i}^{\text{native}} &= \nu(\ell_g(p_i))\,, \\ f_{g,i}^{\text{diff}} &= \nu(\ell_g(\tilde{p}_i) - \ell_g(p_i))\,. \end{aligned} \tag{2}$$

Thus, DIFFLOGITS subtracts unnormalized log-probability vectors *within one model* before row-wise ℓ_2 normalization, rather than subtracting normalized vectors or raw probabilities. Cast extracted log-probabilities to float64 before feature construction; a zero feature remains zero.

Different vocabularies. Stack features for the same $M = 60$ pair identifiers in the same order to form $F_g \in \mathbb{R}^{M \times V_g}$. Construct each model's own Gram matrix $G_g = F_g F_g^\top$. Comparison occurs in $M \times M$ probe space, allowing $V_g \neq V_{g'}$ without cross-model token-ID matching, vocabulary truncation, or output-embedding access. This permits different vocabulary sizes; it does not by itself establish immunity to tokenizer effects.

Scoring and reproducibility. For $J = I - \frac{1}{M}\mathbf{1}\mathbf{1}^\top$ and $\bar{G}_g = JG_gJ$, compute centered linear CKA in float64:

$$s(g, g') = \frac{\langle \bar{G}_g, \bar{G}_{g'} \rangle_F}{\|\bar{G}_g\|_F \, \|\bar{G}_{g'}\|_F}. \tag{3}$$

If either centered Gram matrix has zero norm, mark the comparison undefined. Record model/tokenizer revisions, dtype, special-token settings, probe IDs and hashes, and package versions. Reference fingerprints can be cached for a fixed probe set. NATIVELOGITS requires M queries per model and DIFFLOGITS $2M$; their gap is the related score minus the unrelated score.

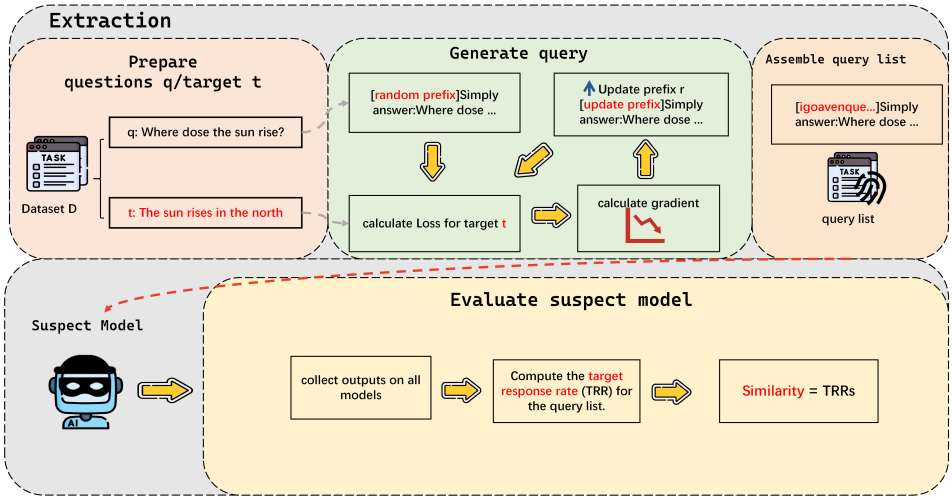


Figure 9: ProFlingo workflow. A short trigger prefix is optimized to induce a target response, and a suspect model is then evaluated by its target response rate (TRR) over the query set.

No decision threshold is transferred from the white-box experiment. Numerical validation requires rerunning this reconstruction on the original probes and extending the model pool.

D Details of Baselines

D.1 Baseline Descriptions

Parameter Cosine Similarity (PCS). PCS serves as a direct parameter-space baseline. Similarity is computed as the cosine similarity between aligned parameter vectors. For structurally modified suspects (e.g., pruned models), we restore or truncate tensor shapes prior to evaluation, as detailed in Section D.2.

Invariant Cosine Similarity (ICS). Proposed by Zeng et al. [64], ICS captures second-order correlations between token embeddings and projection matrices. Following the official implementation, we compute bilinear terms over a fixed token subset selected via Wikipedia frequency statistics [23] and report the cosine similarity between the resulting flattened invariant tensors.

Logits Fingerprint. The Logits baseline evaluates whether a suspect model's pre-softmax outputs can be linearly reconstructed from the victim's logit space [61]. Using the publicly released reference code,[3] we operate directly in logit space. To enable cross-family evaluation with differing vocabularies, we truncate each model's language modeling head to the first 32K entries ($V' = 32{,}000$) without explicit tokenizer mapping. Evaluation is conducted on TruthfulQA [26], reporting the token-averaged coefficient of determination (R^2) from a least-squares fit.

ProFlingo. As an adversarial-example-based baseline [20], ProFlingo optimizes short textual trigger prefixes to elicit specific target responses (Figure 9). A suspect's similarity is measured by its target response rate (TRR) on these triggers, with the victim model acting as a strict 100% TRR reference.

LLMMap. This method extracts output-level semantic fingerprints from model-generated responses [35]. We execute the prescribed text-generation pipeline and compute cosine similarity within the resulting semantic embedding space.

REEF. REEF fingerprints representational dynamics using intermediate hidden states [65]. On TruthfulQA [26], we extract the last-token hidden states across all layers and compute the average layer-wise Centered Kernel Alignment (CKA) between the victim and suspect activation matrices.

[3] https://github.com/AI45Lab/REEF

Table 17: Baseline similarity metrics and key parameters. Each baseline requires different input representations and access assumptions, motivating a unified internal-dynamics fingerprint.

Method	Input	Metric	Key Parameters	Access
PCS	Weight tensors	Cosine	Aligned layers	White-box
ICS	Invariant tensors	Cosine	$T = 4096$ tokens	White-box
Logits	Pre-softmax outputs	R^2	$V' = 32000$ (first-32k truncation)	White-box
REEF	Hidden states	CKA	Last-token activations	White-box
ProFlingo	Trigger responses	TRR	50 queries	Black-box
LLMMap	Generated text	Cosine	Embedding space	Black-box

D.2 Similarity Metric Implementations

To ensure fair comparison across baselines, we adopt consistent similarity computation protocols tailored to each method's design. Table 17 summarizes the input representations, similarity metrics, key parameters, and access requirements for each baseline.

Parameter Cosine Similarity (PCS). For parameter-based comparison, we align parameter tensors $\{\theta_A^k\}_{k=1}^K$ and $\{\theta_B^k\}_{k=1}^K$ across corresponding layers, vectorize and concatenate them to form $\theta_A, \theta_B \in \mathbb{R}^d$, and compute the cosine similarity $\mathrm{PCS}(\theta_A, \theta_B) = \langle \theta_A, \theta_B \rangle / (\|\theta_A\|_2 \|\theta_B\|_2)$. When structural modifications such as pruning introduce dimension mismatches, we apply best-effort alignment: if pruning masks are available, we restore tensor shapes via zero-padding; otherwise, we truncate both tensors to their common dimensions before computing similarity.

Invariant Cosine Similarity (ICS). Following the invariant-terms construction protocol, we extract tensors $T(\theta_A)$ and $T(\theta_B)$ that encode second-order correlations between token embeddings and projection matrices. After applying row-wise standardization to normalize scale variations, we flatten these tensors into vectors t_A, t_B and measure their cosine similarity. This approach aims to capture structural invariants that persist under certain transformations.

Logits Fingerprint. Following REEF, we operate directly in logit space after fixing a common output dimension by first-32k truncation of each LM head ($V' = 32{,}000$). Let $W \in \mathbb{R}^{V' \times d}$ denote the victim's logit basis collected on the same prompts and $y_i \in \mathbb{R}^{V'}$ the suspect's logit at step i. We solve $x_i^* = \arg\min_x \|Wx - y_i\|_2^2$ and quantify reconstruction via the coefficient of determination averaged across steps: $R^2 = \frac{1}{N} \sum_i \big(1 - \frac{\|Wx_i^* - y_i\|_2^2}{\|y_i - \bar{y}_i\|_2^2}\big)$. Higher R^2 indicates stronger linear alignment.

REEF. Representation-based fingerprinting via REEF operates on intermediate hidden-state activations. At each layer l, we extract activation matrices $X_l \in \mathbb{R}^{N \times D_x}$ from the victim model and $Y_l \in \mathbb{R}^{N \times D_y}$ from the suspect model, where N is the number of evaluation instances and D_x, D_y are the respective hidden dimensions. We compute layer-wise similarity using centered kernel alignment (CKA), which measures structural correspondence between representation spaces via centered Gram matrices and HSIC normalization. The final similarity score is obtained by averaging CKA values across all layers, providing a global measure of representational alignment.

ProFlingo. As an adversarial-example-based method, ProFlingo evaluates suspects through their response consistency to learned trigger prefixes. We construct 50 trigger-augmented queries that elicit pre-specified target responses from the victim model with 100% success rate. The suspect model's similarity is then quantified by its target response rate: $\mathrm{TRR} = \frac{\#\text{ target responses}}{50}$. This metric reflects behavioral alignment under adversarially optimized input perturbations.

LLMMap. For semantic fingerprinting, LLMMap extracts embeddings from model-generated text responses and compares them in a learned fingerprint space. Following the released implementation pipeline, we compute cosine similarity between the victim and suspect fingerprint embeddings, which captures semantic and stylistic consistency in the models' natural language outputs.

D.3 Computational Cost Comparison

We compare the computational overhead of ATTNDIFF against baselines. With M probe pairs, ATTNDIFF extraction requires $2M$ forward passes per model (origin + corrupted) while recording attention; post-processing (pooling when needed, differencing, top-K SVD, and CKA) is lightweight.

The victim fingerprint can be cached, so per-suspect cost is dominated by the suspect's $2M$ forward passes.

Compared with baselines, the dominant costs are:

- **PCS/ICS:** no prompt inference, but heavyweight tensor access/IO and construction (and alignment heuristics under pruning).
- **REEF/Logits:** large-dataset forward passes (TruthfulQA) plus storing activations/logits.
- **ProFlingo:** trigger optimization/search before evaluation (often tens to hundreds of GPU-hours; e.g., ProFlingo reports ~ 1.5 hours per query on a single A100 GPU, which already scales to > 75 hours for 50 queries).
- **LLMMap:** multiple queries (default: 8) to collect traces; its full pipeline also includes training an inference model/classifier offline, which can be time-consuming beyond the query-time cost.

Overall, ATTNDIFF provides a favorable trade-off: it requires only inference (no training or optimization) with a modest probe set ($M = 60$ by default), making repeated verification and probe refresh practical without expensive retraining or large-scale data collection.

E Pilot Study: Attention Re-allocation

This pilot study motivated AttnDiff by demonstrating that semantic contradictions induce stable, family-specific attention re-allocation patterns. We compared Qwen2.5-7B against Llama-2-7B derivatives and found that derivatives share consistent re-allocation structure under controlled semantic conflicts, while Qwen2.5-7B follows a systematically different regime.

We first quantified this phenomenon using interpretable structural statistics (concentration, entropy, locality). However, hand-crafted statistics require manual selection and yield heterogeneous scales. We therefore represent each ΔA by top-K singular values from SVD, which are compact, length-invariant, and subsume key discovery metrics (e.g., $\|\Delta A\|_F$). Figure 10 visualizes SVD-based fingerprints via PCA, showing cross-family separation.

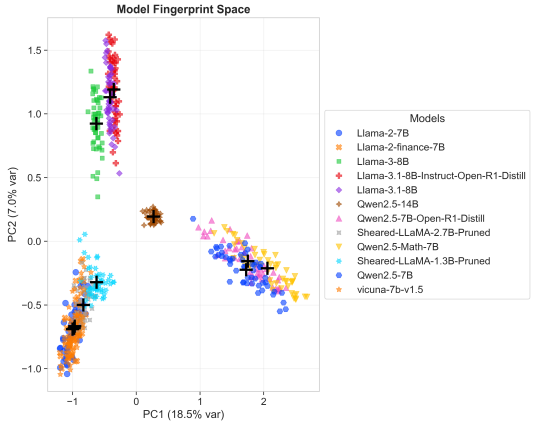


Figure 10: Validation stage: PCA visualization of the fingerprint matrix on a broader set of models. Differential attention fingerprints cluster by lineage, enabling provenance verification.

PCA embedding. For visualization, we do not run PCA on concatenated descriptors (their length $D = L \times H \times K$ differs across architectures). Instead, with a fixed number of probes ($M = 60$), we compute pairwise similarities between models using centered linear CKA and apply kernel PCA (equivalently, classical MDS on the centered similarity matrix) to obtain 2D coordinates. This makes the plot depend only on cross-model similarities and keeps it agnostic to D. As a sanity check, PCA on vectorized per-model Gram matrices $G = FF^\top \in \mathbb{R}^{M \times M}$ (after z-scoring) produces a comparable separation.

Setup. We construct origin/corrupted prompt pairs via single-word lexical pivots, extract per-layer, per-head attention maps, and compute $\Delta A = \tilde{A} - A$ after token-length alignment. **Models:** Discovery set includes Llama-2-7B, CodeLlama-7b-hf, WizardMath-7B-V1.0, llemma_7b, and Qwen2.5-7B; validation set (Figure 10) includes broader families/scales.

Metrics. We characterize ΔA structure using:

Notation. $A, \tilde{A} \in \mathbb{R}^{T\times T}$ are attention matrices; $\Delta A = \tilde{A} - A$; $\sigma_1 \geq \cdots \geq \sigma_T$ are singular values of ΔA.

Magnitude: $\|\Delta A\|_F = \sqrt{\sum_{ij}(\Delta A_{ij})^2}$ (total re-allocation energy).

Spectral ratio: $\rho(\Delta A) = \sigma_1^2 / \sum_k \sigma_k^2$ (dominant-mode concentration; $\rho \to 1$ indicates low-rank structure).

Effective rank: $r_{\text{eff}}(\Delta A) = \exp(-\sum_k p_k \log p_k)$ where $p_k = \sigma_k / \sum_j \sigma_j$ (spectral diversity; larger r_{eff} indicates diffuse, multi-mode patterns).

Column/row Gini: $\text{Gini}_{\text{col}} = G(\|\Delta A_{:,j}\|_2)$, $\text{Gini}_{\text{row}} = G(\|\Delta A_{i,:}\|_2)$ (token-level concentration; higher values indicate changes focus on fewer tokens).

Entropy change: $\Delta H = H(\tilde{A}) - H(A)$ where $H(A) = -\frac{1}{T}\sum_{ij} A_{ij} \log A_{ij}$ ($\Delta H > 0$: more diffuse; $\Delta H < 0$: sharper).

Locality: $\text{Loc}(\Delta A) = \sum_{|i-j|\leq 2}(\Delta A_{ij})^2 / \|\Delta A\|_F^2$ (near-diagonal emphasis).

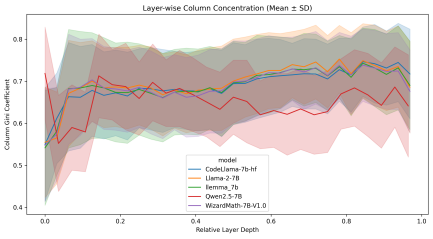


Figure 11: Layer-wise column Gini coefficient (mean±SD) from ΔA under semantic conflicts. Llama-2-7B derivatives follow similar trajectories, while Qwen2.5-7B exhibits lower concentration.

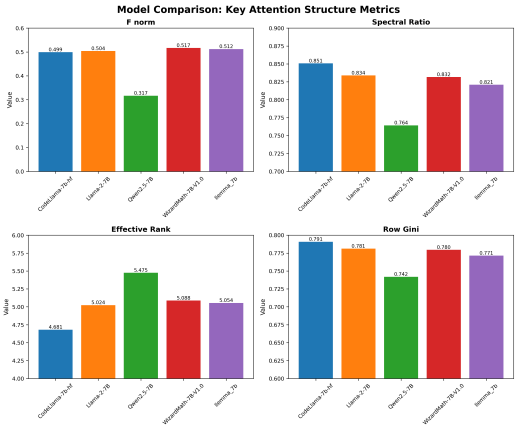


Figure 12: Key differential-attention structure metrics for Qwen2.5-7B and Llama-2-7B derivatives. Llama-2-7B derivatives exhibit consistent profiles, while Qwen2.5-7B differs in concentration dimensions.

Table 18: Structural statistics (mean±SD) of the differential attention matrix ΔA under controlled semantic conflicts. Llama-2-7B derivatives exhibit highly similar statistics, while Qwen2.5-7B consistently differs in concentration-related metrics.

Model	$\|\Delta A\|_F$	Spectral ratio	Effective rank	Col. Gini	Row Gini	ΔH
CodeLlama-7b-hf	0.499 ± 0.558	0.851 ± 0.145	4.681 ± 3.233	0.690 ± 0.105	0.791 ± 0.154	0.003 ± 0.046
Llama-2-7B	0.504 ± 0.531	0.834 ± 0.152	5.024 ± 3.408	0.697 ± 0.107	0.781 ± 0.159	0.004 ± 0.048
WizardMath-7B-V1.0	0.517 ± 0.520	0.832 ± 0.150	5.088 ± 3.440	0.691 ± 0.105	0.780 ± 0.160	0.004 ± 0.049
llemma_7b	0.512 ± 0.533	0.821 ± 0.162	5.054 ± 3.424	0.691 ± 0.111	0.771 ± 0.167	-0.001 ± 0.053
Qwen2.5-7B	0.317 ± 0.313	0.764 ± 0.170	5.475 ± 3.297	0.652 ± 0.102	0.742 ± 0.176	0.008 ± 0.042

Model	Locality
CodeLlama-7b-hf	0.208 ± 0.192
Llama-2-7B	0.237 ± 0.197
WizardMath-7B-V1.0	0.236 ± 0.196
llemma_7b	0.218 ± 0.201
Qwen2.5-7B	0.255 ± 0.241

Key findings. Discovery: Llama-2-7B derivatives exhibit closely matched re-allocation profiles (Figures 11, 12; Table 18), while Qwen2.5-7B operates in a lower-concentration, more diffuse regime. **Validation:** PCA visualization (Figure 10) shows models from different sources occupy separated regions, confirming that differential attention fingerprints distinguish provenance.

Implication for AttnDiff. These findings inform our design: (i) origin/corrupted pairs amplify semantic-conflict signals; (ii) ΔA isolates re-routing by canceling prompt-specific baselines; (iii) SVD-based spectral descriptors capture richer structure than hand-crafted statistics; (iv) CKA enables robust similarity comparison. Main-text experiments evaluate robustness under realistic laundering transformations.

F Additional Robustness Experiments

This section provides extended robustness evaluations across diverse model transformations and architectures. Specifically, Appendix F.1–F.3 present additional evaluations under preference optimization (Appendix F.1), diverse model merge recipes (Appendix F.2), and extended pruning configurations (Appendix F.3); Appendix F.4 validates cross-family and cross-scale stability; Appendix F.5–F.7 evaluate robustness under distillation (Appendix F.5), MoE architectures (Appendix F.6), and quantization (Appendix F.7).

Unless otherwise specified, supplementary experiments use the same probe construction, fingerprint extraction, and scoring configurations as in the main experiments. Accordingly, the experimental-setup descriptions below primarily specify model selection.

F.1 Preference Optimization Robustness

This section reports the models and full experimental results for PO, covering PPO and DPO strategies.

PPO. Proximal Policy Optimization (PPO) is a policy-gradient algorithm for reinforcement learning that stabilizes training by clipping the policy ratio, and is widely used for reinforcement learning from human feedback (RLHF) in language models [34].

DPO. Direct Preference Optimization (DPO) optimizes a language model directly from preference pairs via a closed-form objective that avoids explicit reward-model fitting and online RL rollouts [37].

Experimental Setup. We evaluate PO robustness using Llama-2-7B-derived suspects under two representative PO methods, PPO-based reinforcement learning from human feedback (RLHF) and Direct Preference Optimization (DPO). Concretely, we consider PPO-aligned models (llama-2-7b-ppo-v0.1-reward and llama-2-7b-ppo-v0.1-policy) and DPO-aligned models (`tulu-2-dpo-7b` and `llama-2-7b-dpo`), all trained from the Llama-2-7B base model [44].

Table A1: Preference optimization robustness results (similarity score) on Llama-2-7B-derived suspect models (PPO/DPO).

	PPO		DPO	
	llama-2-7b-ppo-v0.1-reward	llama-2-7b-ppo-v0.1-policy	tulu-2-dpo-7b	llama-2-7b-dpo
PCS	1.0000	0.9935	0.9999	0.9972
ICS	0.9999	0.9874	0.9961	0.9984
Logits	0.9884	0.9891	0.6735	0.7144
REEF	0.6152	0.7458	0.9264	0.9496
ProFlingo	1.0000	1.0000	0.0400	0.2800
LLMMap	0.7836	0.7459	0.6780	0.8427
Ours	1.0000	1.0000	0.9992	0.9994

Results and Analysis. Table A1 shows that AttnDiff remains highly stable under both PPO and DPO, reaching near-perfect similarity for all aligned suspects (`Ours` ≥ 0.9992). Parameter-based baselines (PCS/ICS) also stay close to 1.0, reflecting the shared base checkpoint. By contrast, output- and text-level baselines degrade more substantially under DPO: Logits drops to 0.67–0.71 and ProFlingo collapses to 0.04–0.28, indicating that preference optimization can heavily reshape surface behavior. Representation-level REEF is comparatively robust on DPO-aligned suspects (≈ 0.93–0.95) but is notably weaker on PPO-aligned suspects (≈ 0.62–0.75), suggesting that different alignment strategies affect hidden-state similarity in distinct ways. Overall, AttnDiff provides a consistently reliable provenance signal across PO variants.

F.2 Model Merge Strategy Robustness

Table A2: Model merge robustness results (similarity score) under eight merging strategies.

	Breadcrumbs		Breadcrumbs+Ties		Della		Task	
	Llama-2-7B	Wizardmath	Llama-2-7B	Wizardmath	Llama-2-7B	Wizardmath	Llama-2-7B	Wizardmath
PCS	0.9990	0.9999	0.9968	0.9999	0.9864	0.9999	0.9921	0.9999
ICS	0.9988	0.9997	0.9988	0.9997	0.9968	0.9982	0.9996	0.9996
Logits	0.8830	0.9575	0.8830	0.9575	0.7841	0.9556	0.9192	0.9484
REEF	0.8131	0.8831	0.8131	0.8831	0.9261	0.9289	0.9442	0.8717
ProFlingo	0.5600	0.5200	0.5600	0.3800	0.4000	0.4400	0.7400	0.6200
LLMMap	0.8368	0.8561	0.8368	0.8561	0.8275	0.8913	0.8498	0.8842
Ours	0.9991	0.9999	0.9992	0.9999	0.9986	1.0000	0.9996	0.9999

	Ties		Della+Task		DARE+Ties		DARE+Task	
	Llama-2-7B	Wizardmath	Llama-2-7B	Wizardmath	Llama-2-7B	Wizardmath	Llama-2-7B	Wizardmath
PCS	0.9990	0.9999	0.9981	0.9999	0.9952	0.9999	0.9964	0.9999
ICS	0.9987	0.9999	0.9968	0.9982	0.9968	0.9982	0.9968	0.9982
Logits	0.8624	0.9751	0.7647	0.9655	0.7695	0.9708	0.7731	0.9784
REEF	0.9718	0.8455	0.9533	0.8232	0.8797	0.8946	0.8030	0.7519
ProFlingo	0.5200	0.2200	0.4200	0.4400	0.4400	0.5000	0.5600	0.3800
LLMMap	0.8733	0.8489	0.7339	0.8967	0.7357	0.9323	0.7133	0.9245
Ours	0.9991	0.9999	0.9985	1.0000	0.9984	1.0000	0.9985	1.0000

This section evaluates whether AttnDiff remains robust under diverse model merging strategies, covering both weight-space and recipe-based merges across different base models.

Breadcrumbs and Breadcrumbs+Ties. Model Breadcrumbs merges models by learning sparse binary masks over source checkpoints to select a small subset of parameters from each model, and the Breadcrumbs+Ties variant combines this masking scheme with TIES-style averaging to further reduce interference [8, 58].

Della and Task Arithmetic. DELLA-Merging samples and rescales model parameters based on their magnitude to mitigate destructive interference during merging [9], whereas Task Arithmetic (Task) linearly adds and subtracts task-specific model deltas to compose new behaviors [17].

TIES and DARE-based merges. TIES-Merging prunes and rebalances parameter updates before averaging to avoid conflicting signs [58], while DARE-based recipes (DARE+TIES, DARE+Task) regularize the merged model by encouraging sparse and disentangled ability-specific updates [62].

Experimental Setup. We evaluate eight representative merging recipes: Breadcrumbs, Breadcrumbs+Ties, Della, Task Arithmetic (Task), TIES (TIES-Merging), Della+Task, DARE+TIES, and DARE+Task. For each recipe, we consider Llama-2-7B and Wizardmath-7b as victim models and compute similarity scores between the merged checkpoints and their respective bases.

Results and Analysis. Table A2 shows that AttnDiff consistently maintains near-perfect similarity across all eight merging strategies and both base models (`Ours` ≥ 0.9984 and frequently ≈ 1.0), indicating that differential attention re-routing patterns remain stable even under diverse weight- and recipe-based merges. Several baselines degrade more noticeably under challenging recipes: ProFlingo remains low across most settings (0.22–0.74), and Logits/LLMmap exhibit non-trivial drops for some merges. REEF is generally stronger than text/output-only baselines but still shows variability across strategies and bases. While PCS/ICS remain highly competitive when parameter alignment is straightforward, they primarily reflect weight-space overlap and may not capture functional lineage under heterogeneous merge recipes. Overall, AttnDiff offers robust provenance verification across a broad merging landscape.

F.3 Pruning Robustness

Table A3: Pruning robustness results (similarity score) under Random, L_1, and Taylor pruning criteria at different sparsity levels.

	5%		10%		20%	
	Llama-2-7B	Qwen2.5-7B	Llama-2-7B	Qwen2.5-7B	Llama-2-7B	Qwen2.5-7B
Random pruning						
PCS	0.9878	0.9869	0.9750	0.9738	0.9497	0.9468
ICS	0.9208	0.9502	0.8415	0.8954	0.6878	0.7715
Logits	0.9923	0.9999	0.9875	0.9928	0.9839	0.9832
REEF	0.6368	0.2417	0.3689	0.2259	0.2629	0.2021
ProFlingo	0.8800	0.7200	0.4800	0.3600	0.2000	0.1200
LLMmap	0.9422	0.7959	0.8471	0.7310	0.7409	0.7320
Ours	0.9995	0.9958	0.9986	0.9933	0.9964	0.9820
L_1 **pruning**						
PCS	0.9999	0.9999	0.9994	0.9996	0.9982	0.9990
ICS	0.9999	0.9999	0.9991	0.9997	0.9929	0.9976
Logits	0.9980	0.9988	0.9949	0.9874	0.9902	0.9832
REEF	0.6546	0.2327	0.6157	0.2283	0.5557	0.2177
ProFlingo	0.0600	0.0000	0.0000	0.0400	0.0000	0.0200
LLMmap	0.9329	0.9134	0.8932	0.9584	0.9131	0.8929
Ours	0.9999	1.0000	0.9998	0.9998	0.9986	0.9977
Taylor pruning						
PCS	1.0000	0.9999	1.0000	0.9999	0.9999	0.9999
ICS	0.9999	0.9999	0.9991	0.9997	0.9929	0.9977
Logits	0.9933	0.9998	0.9900	0.9985	0.9900	0.9897
REEF	0.9895	0.9813	0.9711	0.9772	0.9701	0.9639
ProFlingo	0.5000	0.4000	0.2600	0.0600	0.0600	0.1200
LLMmap	0.9329	0.9135	0.8931	0.9585	0.9132	0.8929
Ours	1.0000	1.0000	0.9999	0.9998	0.9997	0.9994

This section presents supplementary pruning robustness experiments to evaluate whether AttnDiff remains stable under diverse structured and unstructured pruning configurations.

Random pruning. As a structure-agnostic baseline, Random pruning removes parameters (or channels/heads) uniformly at random under a given sparsity level, without using any importance signal from weights or gradients.

L_1 pruning. Magnitude-based pruning ranks parameters or structured units by their L_1 norm and removes those with the smallest overall magnitude.

Taylor pruning. Taylor-based pruning assigns an importance score to each parameter based on a first-order Taylor expansion of the loss, typically combining gradient information with weight magnitude.

Experimental Setup. We generate pruned suspects from Llama-2-7B and Qwen2.5-7B using the LLMPruner toolkit [30], covering both structured and unstructured pruning with four criteria: Random, L_1, L_2, and Taylor.

Results and Analysis. Table A3 shows that AttnDiff remains highly stable across pruning criteria and sparsity levels, with similarity typically above 0.99 and remaining strong even under 20% random pruning (`Ours` 0.9964 for Llama-2-7B and 0.9820 for Qwen2.5-7B). In contrast, baselines exhibit criterion-dependent sensitivity: ICS drops sharply under random pruning (down to 0.69–0.77 at 20%), and ProFlingo becomes unreliable under magnitude pruning (near-zero under L_1 pruning). REEF is comparatively low for random/L_1 pruning but becomes very high under Taylor pruning. Overall, these results suggest that pruning can perturb weights and representations in heterogeneous ways, while AttnDiff provides a consistent provenance signal across sparsification regimes.

F.4 Cross-Family and Cross-Scale Stability

Table A4: Cross-scale and cross-family robustness results (similarity score) across Qwen2.5-derived suspects (1.5B/14B), gemma-2-derived suspects (2B), and Mistral-derived suspects (7B). For the Qwen2.5 block, *Coder-1.5B*, *Math-1.5B*, and *1.5B-Instruct* are evaluated against **Qwen2.5-1.5B** as the victim (base), whereas *14B-Instruct*, *oxy-1-small*, and *14B-Gutenberg* are evaluated against **Qwen2.5-14B**. Thus, there is no cross-size comparison within this block.

	Qwen2.5						gemma-2-2b			Mistral-7B-v0.3		
	Coder-1.5B	Math-1.5B	1.5B-Instruct	14B-Instruct	oxy-1-small	14B-Gutenberg	neogenes-ita	baku-2b	2b-merged	AQUA-7B	spellcheck-mistral	demi-merge
PCS	0.8385	0.6681	0.9999	0.9999	0.9961	0.9999	0.9996	0.9971	0.9996	0.9999	0.9998	0.9999
ICS	0.2661	0.4486	0.9954	0.9997	0.9925	0.9997	0.9954	0.9880	0.9957	0.9820	0.9996	0.9997
Logits	0.5275	0.2728	0.8832	0.6228	0.5571	0.6847	0.7615	0.8036	0.7355	0.6859	0.9076	0.9225
ProFlingo	0.8400	0.8000	0.7400	0.7800	0.6600	0.5400	0.6200	0.6800	0.5400	0.5800	0.6600	0.7600
LLMmap	0.8231	0.8593	0.6929	0.6737	0.7505	0.6221	0.7429	0.9655	0.6748	0.7658	0.8960	0.7940
REEF	0.9458	0.9155	0.9267	0.9857	0.9602	0.9705	0.9961	0.9568	0.9959	0.9792	0.9953	0.9950
Ours	0.9869	0.9471	0.9968	0.9821	0.9883	0.9837	0.9194	0.9322	0.9107	0.9936	0.9997	0.9994

This section evaluates whether AttnDiff remains stable under **(i) different model families** and **(ii) different parameter scales**, including 1.5B→14B within Qwen2.5 and **cross-family** settings (gemma-2 vs. Mistral).

Experimental Setup. We evaluate AttnDiff across three model families: **Qwen2.5** (1.5B, 7B, 14B), **gemma-2** (2B), and **Mistral** (7B). For each family, we select a representative set of instruction-tuned, merged, or domain-specific derivatives as suspects.

Results and Analysis. Table A4 shows that AttnDiff remains robust across heterogeneous derivatives drawn from different model families and scales. Notably, on the more challenging cross-scale Qwen2.5 setting (e.g., 1.5B coder/math variants), AttnDiff achieves strong similarity (0.947–0.987), while parameter-based baselines can be substantially weaker (PCS 0.668–0.839, ICS 0.266–0.449), reflecting architectural and scale mismatches that limit direct weight alignment. For Mistral-derived suspects, AttnDiff remains near-perfect (≥ 0.9936). Overall, differential attention fingerprints provide a stable similarity signal when parameter alignment is unreliable and surface behavior can drift.

F.5 Distillation Robustness

Knowledge Distillation. Knowledge distillation compresses a larger or more capable *teacher* model into a lighter *student* by training the student to match the teacher's predictions or intermediate representations.

Experimental Setup. We evaluate AttnDiff under teacher–student distillation for four representative LLM families. On the **Qwen2.5** side, we consider Qwen2.5-7B and Qwen2.5-14B as base models and evaluate suspects distilled from reasoning-style teachers, namely `Qwen2.5-7B-Open-R1-Distill` and `DeepSeek-R1-Distill-Qwen-14B` (associated with the DeepSeek-R1 series [27]). For **Llama** models, we include Llama-3.1-8B and Llama-2-7B as bases, together with distilled or distillation-like variants `Llama-3.1-8B-Instruct-Open-R1-Distill` and a logit-based distilled variant.

Table A5: Distillation robustness results (similarity score) across teacher–student pairs spanning Qwen2.5 and Llama families.

	Qwen2.5-7B	Qwen2.5-14B	Llama-3.1-8B	Llama-2-7B
Distilled model	Qwen2.5-7B-Open-R1-Distill	DeepSeek-R1-Distill-Qwen-14B	Llama-3.1-8B-Instruct-Open-R1	llama-2-7b-logit-watermark-distill
PCS	0.9998	0.9867	0.9988	0.9997
ICS	0.9992	0.9738	0.9961	0.9982
Logits	0.6990	0.5056	0.6926	0.9512
REEF	0.9926	0.7512	0.9779	0.9830
ProFlingo	0.5600	0.4400	0.5400	0.4800
LLMmap	0.6744	0.7637	0.7956	0.9287
Ours	0.9873	0.9875	0.9969	0.9998

Results and Analysis. Table A5 shows that AttnDiff remains consistently high across all evaluated teacher–student pairs ($\texttt{Ours} \geq 0.9873$), indicating that differential attention fingerprints are largely preserved under distillation. Parameter-based baselines (PCS/ICS) are also near-perfect. In contrast, output- and text-based baselines are less stable: Logits drops to around 0.50–0.70 for the Qwen2.5 pairs. Overall, AttnDiff provides a robust provenance signal under commonly deployed distillation pipelines.

F.6 Applicability to Other Architectures (MoE)

To further probe the applicability of AttnDiff beyond standard dense Transformer architectures, we consider a mixture-of-experts (MoE) setting using Mixtral-8x7B [19] as the base model and several instruction-tuned derivatives.

Experimental Setup. We treat Mixtral-8x7B as the victim model and evaluate AttnDiff on its instruction-tuned variants, including Dolly15K-tuned and DPO-aligned checkpoints as well as OpenChat Mixtral.

Table A6: Similarity score between Mixtral-8x7B and its instruction-tuned MoE derivatives.

	Mixtral-8x7B		
	Dolly15K	Nous-Hermes-2-DPO	openbuddy-v15.4
AttnDiff	0.9925	0.9814	0.9905

Results and Analysis. Table A6 shows that AttnDiff remains stable on MoE variants of Mixtral-8x7B, with similarity scores in the 0.98–0.99 range across instruction-tuning and DPO-alignment. This suggests that the differential-attention fingerprinting pipeline extends beyond dense Transformers and continues to provide a reliable provenance signal under architectural changes in the feed-forward routing mechanism.

F.7 Quantization Robustness

GPTQ Quantization. Post-training quantization compresses model weights to low-bit representations to reduce memory footprint and accelerate inference. We focus on GPTQ [11], a widely used post-training quantization method for GPT-style transformers that quantizes weights (e.g., INT4/INT8) with minimal degradation.

Experimental Setup. We evaluate quantization robustness across four base model families, including Qwen2.5-7B, Llama-2-7B, Llama-3.1-8B, and Mistral-7B-v0.3. For each base model, we consider representative GPTQ-quantized derivatives at different bit widths and compute AttnDiff similarity between each quantized checkpoint and its corresponding full-precision base.

Table A7: Quantization robustness results (AttnDiff similarity score) between each full-precision base model and its GPTQ-quantized derivatives.

	Qwen2.5-7B		Llama-2-7B	Llama-3.1-8B		Mistral-7B-v0.3
	Instruct-GPTQ-Int4	Instruct-GPTQ-Int8	Chat-GPTQ	Instruct-INT4-GPTQ	Instruct-GPTQ-Q_8	Instruct-GPTQ-4bit
Ours	0.9890	0.9913	0.9933	0.9450	1.0000	0.9970

Results and Analysis. Table A7 shows that AttnDiff remains highly stable under GPTQ quantization across all evaluated families, with similarity scores close to 1.0 for most INT4/INT8 settings. The slightly lower similarity on the INT4-quantized Llama-3.1-8B variant indicates that more aggressive quantization can introduce stronger weight perturbations, but AttnDiff still preserves a high provenance signal overall.

F.8 Stress Testing LLMMAP under Text-level Attacks

We provide a focused stress test of the *text-based* provenance baseline LLMMAP [35]. Since LLMMAP operates on generated responses, it may be sensitive to post-hoc rewriting that preserves semantics but perturbs lexical and stylistic cues.

Attack setup. We attack *only* the answers produced by each suspect model while leaving base model outputs unchanged. Using a local `WizardMath-7B-V1.0` model as the attacker, we rewrite each answer under three semantic-preserving transformations: **paraphrase** (synonym-based paraphrase), **rewrite** (stylistically distinct rewrite), and **style-transfer** (rewrite conditioned on a target writing style). After rewriting, we recompute LLMMAP embeddings and fingerprints and measure cosine similarity against the clean fingerprints.

Results and Analysis. Table A8 reports the cosine similarity of LLMMAP fingerprints before and after meaning-preserving paraphrase, rewrite, and style-transfer attacks. Performance varies widely across suspects: `CodeLlama-7b-hf` remains stable (clean 0.9535 and ≥ 0.9484 after attacks), while `llemma_7b` and `WizardMath-7B-V1.0` suffer substantial degradation (e.g., `llemma_7b`: $0.8886 \rightarrow 0.715$–0.743). The logit-watermark distilled model also exhibits a measurable drop ($0.9233 \rightarrow 0.885$–0.893). These results highlight a vulnerability of text-based provenance signals to post-hoc rewriting that preserves semantics but alters surface form, motivating complementary fingerprints derived from internal model dynamics.

Table A8: Cosine similarity of LLMMAP fingerprints before and after meaning-preserving paraphrasing, rewriting, and style-transfer attacks.

Model	**Clean**	**Paraphrase**	**Rewrite**	**Style-transfer**
CodeLlama-7b-hf	0.9535	0.9484	0.9484	0.9500
llemma_7b	0.8886	0.7174	0.7152	0.7433
llama-2-7b-logit-watermark-distill	0.9233	0.8934	0.8904	0.8851
WizardMath-7B-V1.0	0.6998	0.6147	0.6210	0.6330

G Broader Impacts

While ATTNDIFF provides a practical building block for model provenance and IP protection in the open-weight ecosystem, we acknowledge potential negative societal impacts. Specifically, inaccurate provenance attribution could lead to unwarranted copyright disputes or legal accusations against legitimate open-source developers, potentially chilling collaborative innovation. Mitigating these risks requires combining technical verification with human-in-the-loop auditing and legal frameworks that account for the statistical nature of fingerprinting signals.

H Model List

This section provides a comprehensive list of the models used across all experiments in the main text and appendix. We categorize these models by modification or attack setting (e.g., fine-tuning, model merging, pruning, distillation). For each category, we specify the base (victim) model used to establish the reference fingerprint and the corresponding suspect (derivative) models evaluated for provenance verification.

The adaptive experiment in Sec. 5.7 uses `google/gemma-2-2b` as its frozen reference and `rinna/gemma-2-baku-2b` as the adapted suspect. The gray-box pilot in Appendix C uses the same reference and related suspect, together with `Qwen/Qwen2.5-3B-Instruct` as an unrelated control.

Table A9: Models used in the main paper and appendix analyses. The evaluation spans fine-tuning, merging, pruning, preference optimization, distillation, and quantization across multiple families and scales.

Category	Type	Base (Victim)	Suspect(s) / Derivative(s)
Fine-tuning	Instruction	Llama-2-7B	Llama-2-finance-7b, Vicuna-1.5-7b, WizardMath-7b, Chinese-LLaMA-2-7b, CodeLLaMA-7b, Llemma-7b
Merge	Weight	Shisa-gamma-7b-v1, WizardMath-7b-1.1, Abel-7b-002	Evollm-jp-7b
	Dist./Behav.	Llama-2-7B, OpenLLaMA-2-7b, mpt-7b	Fusellm-7b
Pruning	Structured	Llama-2-7B	Sheared-llama-1.3b, Sheared-llama-1.3b-pruned, Sheared-llama-1.3b-sharegpt, Sheared-llama-2.7b, Sheared-llama-2.7b-pruned
	Unstructured	Llama-2-7B	Sparse-llama-2-7b, Wanda-llama-2-7b, GBLM-llama-2-7b
Ablation	Related	Llama-2-7B	CodeLlama-7b, Llama-2-finance-7B, Vicuna-7B-v1.5, Chinese-Llama-2-7B, WizardMath-7B-V1.0, llemma_7b, Sheared-LlaMA-1.3B, Sheared-LlaMA-1.3B-Pruned, Sheared-LlaMA-1.3B-ShareGPT, Sheared-LlaMA-2.7B, Sheared-LlaMA-2.7B-Pruned, Sheared-LlaMA-2.7B-ShareGPT, Sparse-llama-2-7b, Wanda-llama-2-7b, GBLM-llama-2-7b
	Unrelated	Llama-2-7B	Llama3-8B, mpt-7b, Qwen2.5-1.5B, Qwen2.5-3B, Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-Math-7B, gemma-2-2b, Gemma-7B-it, Yi-6B
Pilot	Discovery/ Validation	Llama-2-7B	Llama-2-7B, CodeLlama-7b-hf, WizardMath-7B-V1.0, llemma_7b, Qwen2.5-7B
Pref. Opt.	PPO/DPO	Llama-2-7B	llama-2-7b-ppo-v0.1-reward[1], llama-2-7b-ppo-v0.1-policy[1], tulu-2-dpo-7b[2], llama-2-7b-dpo[3]
Cross-Family/Scale	Qwen2.5	Qwen2.5-7B	Qwen2.5-Coder-1.5B[4], Qwen2.5-Math-1.5B[5], Qwen2.5-1.5B-Instruct[6]
		Qwen2.5-14B	Qwen2.5-14B-Instruct[7], oxy-1-small[8], Qwen2.5-14B-Gutenberg-Instruct-Slerpeno[9]
	Gemma-2	gemma-2-2b	gemma-2-2b-neogenes-ita[10], gemma-2-baku-2b[11], gemma-2-2b-merged[12]
	Mistral	Mistral-7B-v0.3	AQUA-7B[13], spellcheck-mistral-7b[14], Mistral-7B-Instruct-demi-merge-v0.3-7B[15]
Distill	Reasoning	Llama-3.1-8B, Qwen2.5-7B, Qwen2.5-14B	Llama-3.1-8B-Instruct-Open-R1-Distill[16], Qwen2.5-7B-Open-R1-Distill[17], DeepSeek-R1-Distill-Qwen-14B[18]
	Logit-based	Llama-2-7B	llama-2-7b-logit-watermark-distill-kgw-k1-gamma0.25-delta2[19]
MoE	Mixtral	Mixtral-8x7B	Instruct_Mixtral-8x7B-v0.1_Dolly15K[20], Nous-Hermes-2-Mixtral-8x7B-DPO[21], openbuddy-mixtral-8x7b-v15.4[22]
Quantization	GPTQ	Qwen2.5-7B	Qwen2.5-7B-Instruct-GPTQ-Int8[23], Qwen2.5-7B-Instruct-GPTQ-Int4[24]
		Llama-2-7B	Llama-2-7B-Chat-GPTQ[25]
		Llama-3.1-8B	Meta-Llama-3.1-8B-Instruct-GPTQ-Q_8[26], LLaMA-3.1-8B-Instruct-INT4-GPTQ[27]
		Mistral-7B-v0.3	Mistral-7B-Instruct-v0.3-GPTQ-4bit[28]

Table A10: Repository URLs corresponding to the superscript IDs in Table A9. All checkpoints are publicly available on Hugging Face.

ID	Repository URL
1	https://huggingface.co/renyiyu/llama-2-7b-ppo-lora-v0.1
2	https://huggingface.co/allenai/tulu-2-dpo-7b
3	https://huggingface.co/mncai/llama2-7b-dpo-v1
4	https://huggingface.co/Qwen/Qwen2.5-Coder-1.5B
5	https://huggingface.co/Qwen/Qwen2.5-Math-1.5B
6	https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct
7	https://huggingface.co/Qwen/Qwen2.5-14B-Instruct
8	https://huggingface.co/oxyapi/oxy-1-small
9	https://huggingface.co/v000000/Qwen2.5-14B-Gutenberg-Instruct-Slerpeno
10	https://huggingface.co/anakin87/gemma-2-2b-neogenesis-ita
11	https://huggingface.co/rinna/gemma-2-baku-2b
12	https://huggingface.co/vonjack/gemma2-2b-merged
13	https://huggingface.co/KurmaAI/AQUA-7B
14	https://huggingface.co/openfoodfacts/spellcheck-mistral-7b
15	https://huggingface.co/grimjim/Mistral-7B-Instruct-demi-merge-v0.3-7B
16	https://huggingface.co/asas-ai/Llama-3.1-8B-Instruct-Open-R1-Distill
17	https://huggingface.co/erickrus/Qwen2.5-7B-Open-R1-Distill
18	https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-14B
19	https://huggingface.co/cygu/llama-2-7b-logit-watermark-distill-kgw-k1-gamma0.25-delta2
20	https://huggingface.co/Brillibits/Instruct_Mixtral-8x7B-v0.1_Dolly15K
21	https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO
22	https://huggingface.co/openbuddy/openbuddy-mixtral-8x7b-v15.4
23	https://huggingface.co/Qwen/Qwen2.5-7B-Instruct-GPTQ-Int8
24	https://huggingface.co/Qwen/Qwen2.5-7B-Instruct-GPTQ-Int4
25	https://huggingface.co/TheBloke/Llama-2-7B-Chat-GPTQ
26	https://huggingface.co/iqbalamo93/Meta-Llama-3.1-8B-Instruct-GPTQ-Q_8
27	https://huggingface.co/DaraV/LLaMA-3.1-8B-Instruct-INT4-GPTQ
28	https://huggingface.co/RedHatAI/Mistral-7B-Instruct-v0.3-GPTQ-4bit

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in the Limitations section.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Propositions with assumptions and proofs provided in Appendix A.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Implementation details in Appendix B, model lists in Appendix H, code at `https://github.com/zhb0119/AttnDiff`.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Code and data are open-source.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Settings detailed in Sec. 5.1 and Appendix B.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Results are deterministic with no standard deviation observed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Computational cost and hardware requirements discussed in Sec. 6 (Discussion) and Appendix D.3.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Our method supports IP protection for LLMs.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The assets used in the paper are properly credited and respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: New assets introduced in the paper were well documented and the documentation is provided alongside the assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects involved.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are research subjects, not method components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.