FUNCTIONAL NETWORK FINGERPRINT: A UNIFIED METHOD FOR CROSS-ARCHITECTURE AND CROSS-STRUCTURE LLM LINEAGE VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Training large language models (LLMs) is extremely expensive, yet released weights can be copied and re-released by third parties. We study a subtle disguise, which we call *weight repackaging*: an attacker copies a victim model's weights and then alters its structure—adding, removing, or interpolating layers, or expanding the hidden or MLP width—before continuing training. The resulting model differs from the victim in depth, width, and tokenizer while inheriting most of its parameters, making fingerprints that rely on direct weight or representation correspondence difficult to apply. We propose the Functional Network Fingerprint (FNF), a white-box method that requires no fingerprint-specific training and verifies whether a suspect model is derived from a victim model. Inspired by brain-fingerprinting research, FNF applies canonical independent component analysis (CanICA) to internal activations of Transformer blocks to identify co-activating neuron groups, and compares their activity trajectories between two models under identical input stimuli. Because FNF compares functional activity rather than weights or static representations, it does not require direct correspondence in depth, width, or tokenizer, and can operate across architectures and under common weight transformations. On a labeled set of 40 model pairs spanning three provenance categories, FNF separates derived from independent models with an AUC of 1.00 and identifies every derived model at zero false positives. On the 36 pairs to which two non-invasive baselines also apply, the representation-based method REEF reaches an AUC of 0.90 and the recent attention-difference method AttnDiff 0.98, and at zero false positives they recover only 50% and 83% of the related pairs. FNF thus provides a simple and interpretable tool for LLM provenance verification.

1 INTRODUCTION

Training large language models (LLMs) is costly Zhao et al. (2023); Liu et al. (2023b); Wang et al. (2024); Liu et al. (2025b). Although open-source models are released under specific licenses, third parties may copy released weights, modify the resulting model, and redistribute it as a distinct model. Reliable model fingerprints can help owners and third parties verify whether a suspect model is derived from a victim model or independently created.

Existing approaches fall into two broad categories. *Invasive* watermarking methods embed triggers or additional parameters into a model Peng et al. (2023); Xu et al. (2024; 2025), but they may affect model utility and cannot be applied retroactively to already-released models Russinovich & Salem (2024). *Non-invasive* methods instead exploit intrinsic properties of a model, including output-based detectors Mitchell et al. (2023); Sadasivan et al. (2023); Wu et al. (2023); Iourovitski et al. (2024); Pasquini et al. (2025); McGovern et al. (2024), weight-similarity tests Chen et al. (2023; 2022); Zeng et al. (2024), and representation-similarity fingerprints such as REEF Zhang et al. (2025). These approaches are promising, but weight- and representation-based fingerprints generally rely on some form of architectural or representational correspondence: comparing two models typically requires compatible layers, hidden dimensions, or tokenization.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A realistic form of misuse breaks this correspondence. We call it *weight repackaging*: an attacker copies an open-source LLM's weights and then changes its *structure*—duplicating or interpolating layers, expanding the hidden or intermediate dimensions, or deleting layers—before continuing training. The resulting model is structurally different from the victim in depth, width, and potentially tokenizer, while retaining much of its inherited parameters. Weight-based methods become difficult to apply after such dimensional changes, and representation-based methods such as REEF cannot directly align models whose internal representations have different shapes. SOLAR-10.7B, obtained from Mistral-7B by up-scaling its depth from 32 to 48 layers Kim et al. (2024b), provides a concrete example of this type of architectural repackaging.

We propose the *Functional Network Fingerprint* (FNF), inspired by brain-fingerprinting research in neuroscience, where functional brain activity can serve as an individual-specific fingerprint Finn et al. (2015); Van De Ville et al. (2021). We ask whether an analogous signal exists in LLMs. Treating neuron activations within Transformer blocks as functional activity, FNF applies spatial independent component analysis (ICA) Hyvärinen & Oja (2000); Varoquaux et al. (2010b;a) to identify groups of neurons that co-activate (*functional networks*). For models that share a lineage, the activity trajectories of these functional networks can remain consistent across the two models under identical input stimuli, even when their architectures differ; independently trained models exhibit substantially weaker consistency. This functional comparison avoids requiring explicit structural correspondence, allowing FNF to operate across architectures and dimensions.

We evaluate FNF on a range of open-model families and their fine-tuned, pruned, permuted, merged, and repackaged descendants. Across these settings, FNF identifies the evaluated derived models and rejects independently trained models with matched or related architectures. FNF also extends representation-based fingerprinting to weight repackaging, where direct architectural correspondence is unavailable. We further calibrate the continuous FNF score using an independence-based null analysis, a complementary aligned-network count, and ROC-AUC evaluation. On a labeled set of 40 model pairs spanning three provenance categories, FNF achieves an ROC-AUC of 1.00 and detects every derived model without false positives. Recomputing two non-invasive baselines on the 36 pairs to which they also apply gives an AUC of 0.90 for the representation-based method REEF and 0.98 for the attention-difference method AttnDiff Zhang et al. (2026); at zero false positives they recover only 50% and 83% of the related pairs.

2 Related Work

Invasive fingerprinting. Watermarking methods embed information into parameters or behavior Peng et al. (2023); Adi et al. (2018); Zhang et al. (2018); Li et al. (2019); Guo & Potkonjak (2018); Li et al. (2023); Xu et al. (2024) and may degrade quality or be inapplicable post-release Russinovich & Salem (2024).

Non-invasive fingerprinting. Output-based methods fingerprint by text provenance Mitchell et al. (2023); Sadasivan et al. (2023); Wu et al. (2023); Iourovitski et al. (2024); Pasquini et al. (2025); McGovern et al. (2024); weight-similarity methods Chen et al. (2023; 2022); Zeng et al. (2024) are fragile to parameter permutation; REEF Zhang et al. (2025) compares internal representations by CKA and is currently the strongest representation-based baseline able to compare mismatched feature dimensions; AttnDiff Zhang et al. (2026) fingerprints the per-head attention difference between an original and a minimally corrupted prompt. We compare against both; as Appendix E shows, neither cleanly separates repackaged models from same-tokenizer models that share no weights. The black-box alternatives instead require query access to a deployed model.

Functional networks in LLMs. Prior work has used ICA-like decomposition to localize functional networks and key neurons in LLMs for interpretability Liu et al. (2025a). FNF's contribution is not the discovery of functional networks themselves, but the observation that the *consistency of functional-network activity* is a robust fingerprint for lineage verification.

3 Problem Statement and Threat Model

We frame model fingerprinting as *lineage verification*. A victim model $\mathcal{M}_v$ is released under a license, and a suspect model $\mathcal{M}_s$ is provided to a verifier who must decide whether $\mathcal{M}_s$ derives

from $\mathcal{M}_v$, i.e., whether a substantial part of $\mathcal{M}_s$'s computation is inherited from $\mathcal{M}_v$. FNF is a white-box method: the verifier can run forward passes on both models and read their internal activations, which is the natural setting when a suspect model's weights are themselves available. An API-only, black-box verifier is out of scope.

The attacker has broad latitude: they may fine-tune $\mathcal{M}_v$, prune it, permute its parameters, merge it with other models, or apply weight repackaging. We evaluate FNF under all of these transformations in Section 6.

Our focus is *weight repackaging*: the attacker copies $\mathcal{M}_v$'s weights, applies a weight-preserving structural transformation—duplicating or interpolating layers, expanding the hidden or MLP width, or adding/removing layers—and optionally continues training, so the released model differs in architecture but inherits most parameters. Because FNF extracts functional-network activity trajectories independently within each model, it never aligns specific blocks or dimensions, and can therefore handle structural changes such as removing or inserting layers without a one-to-one block mapping Kim et al. (2024a).

4 PRELIMINARIES

Functional brain networks. Functional brain networks (FBNs) are sets of regions that co-activate Bullmore & Sporns (2009); Liu et al. (2023a; 2024b;a); He et al. (2023); Qiang et al. (2022; 2024), e.g. the default-mode network Raichle (2015); Smith et al. (2009). Connectivity-based "brain fingerprints" are well established Finn et al. (2015); Van De Ville et al. (2021). We transfer this concept to LLMs, treating neuron activations as fMRI-like signals.

Independent component analysis. ICA models an observed signal as a linear mixture of statistically independent sources. In fMRI, spatial ICA decomposes a $T \times V$ (time $\times$ voxel) matrix into K independent *spatial maps*, each a functional network. Canonical ICA (CanICA) Varoquaux et al. (2010b;a) first decorrelates and whitens via PCA, then applies FastICA Hyvärinen & Oja (2000); it is a standard, interpretable tool for group-level functional-network extraction. We use it directly on LLM activations.

5 METHOD

5.1 SIGNALS AND SPATIAL DECOMPOSITION

Consider N input sequences processed by a model $\mathcal{M}$. For each sequence n, we collect the hidden states of *all* Transformer blocks and concatenate them along the feature dimension, giving

$$\mathbf{X}^{(n)} \in \mathbb{R}^{T_n \times D}, \qquad D = L \cdot h, \tag{1}$$

where T_n is the length of sequence n, L is the number of Transformer blocks, h is the hidden width, and $D = L \cdot h$ is the total number of "neural" channels. Concatenating all samples:

$$\mathbf{X} = [\mathbf{X}^{(1)}; \ldots; \mathbf{X}^{(N)}] \in \mathbb{R}^{T \times D}, \quad T = \sum_{n=1}^{N} T_n. \tag{2}$$

This concatenation is exactly how FNF integrates layers: each block contributes its h-dimensional hidden state, and the block outputs are stacked to form the full feature/neuron axis D. Because D is defined *per model*, FNF needs no dimension matching between the two models.

Whitening. CanICA first removes second-order correlations and normalizes variance:

$$\mathbf{Z} = \mathbf{X}\,\mathbf{W} \in \mathbb{R}^{T \times K}, \qquad \mathbf{W} \in \mathbb{R}^{D \times K}, \qquad \tfrac{1}{T}\mathbf{Z}^\top \mathbf{Z} = \mathbf{I}_K, \tag{3}$$

where $K \ll D$ is the number of latent components (functional networks). Each row $\mathbf{z}_t$ is the token t's state in a low-dimensional, decorrelated, unit-variance latent space; $\mathbf{W}$ contains the top-K right singular vectors of $\mathbf{X}$ scaled by inverse singular values.

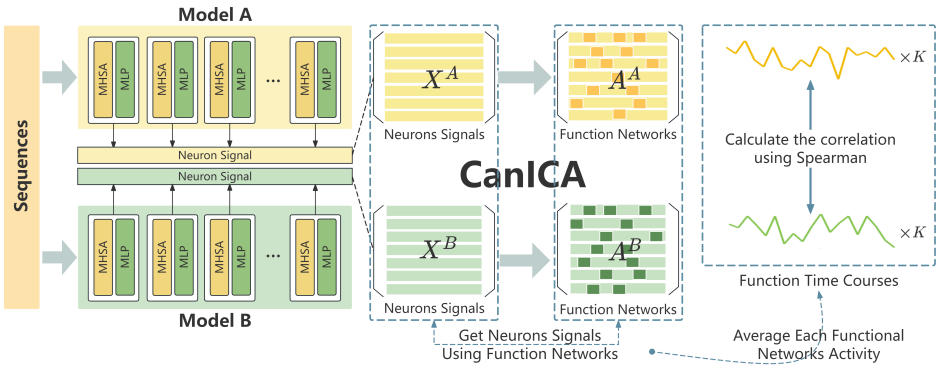


Figure 1: FNF framework. Both models receive identical input stimuli. For each model we extract the hidden states of all blocks, concatenate them into a $T \times D$ signal ($D = L \cdot h$), run CanICA to obtain K spatial maps, threshold each to a core-neuron mask, and record the per-network activity time course. FNF compares the resulting time courses by Spearman correlation, giving a $K \times K$ matrix whose maximum (and aligned-network count) yields the score.

Independent spatial components. CanICA models the whitened data as a linear mixture of *spatially* independent sources. Reverting the whitening, the original signal admits the reconstruction

$$\mathbf{X} \approx \mathbf{S}\,\mathbf{A}^\top, \qquad \mathbf{S} \in \mathbb{R}^{T\times K}, \qquad \mathbf{A} \in \mathbb{R}^{D\times K}, \tag{4}$$

where $\mathbf{S}$ collects the token-level *time courses* (activation dynamics across tokens) and $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_K]$ collects the *spatial maps*, each column $\mathbf{a}_k \in \mathbb{R}^D$ defining one functional network over the D neuron channels. The independent components we use downstream are the spatial maps $\mathbf{A}$.

Thresholding. Following neuroimaging practice, we z-score each spatial map and threshold at $|z| > 3.6$ (a conservative, sparse cutoff; the resulting core networks involve $< 2\%$ of neurons in our experiments), yielding binary masks $\mathbf{m}_k \in \{0,1\}^D$ marking each network's core neurons.

5.2 Functional-network fingerprints and activity time courses

The pipeline is shown in Figure 1. We compute CanICA independently on each model's concatenated signal $\mathbf{X}^{\mathrm{A}}$ and $\mathbf{X}^{\mathrm{B}}$, obtaining K spatial maps each. After thresholding, for each sample n and network i we average the model's hidden states over the masked core neurons to obtain a T_n-dimensional activity time course:

$$\mathbf{s}_i^{\mathrm{A},(n)} = \left[\overline{\mathbf{x}_{t,\bullet}^{\mathrm{A},(n)}}_{\,\mathbf{m}_i^{\mathrm{A}}}\right]_{t=1}^{T_n} \in \mathbb{R}^{T_n}. \tag{5}$$

Cross-tokenizer alignment. The two models may tokenize the same text into different sequence lengths. Rather than requiring token correspondence, FNF compares the *global dynamics* of each network. We uniformly resample each time course to a common length L_{c} (taken as the smaller of the two lengths for a given sample). This is analogous to resampling fMRI signals acquired at different temporal resolutions: it changes only the temporal resolution, not the overall evolution, of the trajectory, and, because adjacent tokens are highly correlated, it preserves the dynamic information that matters for the comparison while making the lengths comparable.

Cross-model similarity. For each sample n, we compute the Spearman rank correlation between every pair of network time courses:

$$\rho_{ij}^{(n)} = \mathrm{Spearman}\left(\mathbf{s}_i^{\mathrm{A},(n)}, \mathbf{s}_j^{\mathrm{B},(n)}\right), \qquad \forall\, i,j \in \{1,\ldots,K\}. \tag{6}$$

Averaging across samples yields the mean consistency matrix

$$\bar{\mathbf{R}} = [\,\bar{\rho}_{ij}\,], \qquad \bar{\rho}_{ij} = \tfrac{1}{N}\sum_{n=1}^{N} \rho_{ij}^{(n)}. \tag{7}$$

Inspired by the fact that Spearman correlation is robust to monotonic transformations (so it tracks trend consistency rather than absolute scale), this is the primary similarity signal.

Algorithm 1 Functional Network Fingerprint (FNF)

Input: Activation sequences $\{\mathbf{X}^{\mathrm{A},(n)}\}_{n=1}^{N}$ and $\{\mathbf{X}^{\mathrm{B},(n)}\}_{n=1}^{N}$, where $\mathbf{X}^{\mathrm{A},(n)}, \mathbf{X}^{\mathrm{B},(n)} \in \mathbb{R}^{T_n \times D}$; number of networks K; common resampled length L_{c}; aligned-network threshold τ_a.
Output: Similarity matrix $\bar{\mathbf{R}} \in \mathbb{R}^{K \times K}$, FNF score, and aligned-network count C.
// Spatial decomposition
$\mathbf{A}^{\mathrm{A}} \leftarrow \textsc{CanICA}(\{\mathbf{X}^{\mathrm{A},(n)}\}_{n=1}^{N}, K)$ ▷ $\mathbf{A}^{\mathrm{A}} \in \mathbb{R}^{D \times K}$
$\mathbf{A}^{\mathrm{B}} \leftarrow \textsc{CanICA}(\{\mathbf{X}^{\mathrm{B},(n)}\}_{n=1}^{N}, K)$ ▷ $\mathbf{A}^{\mathrm{B}} \in \mathbb{R}^{D \times K}$
for $q \in \{\mathrm{A}, \mathrm{B}\}$ **do**
  for $k = 1, \ldots, K$ **do**
    $\mathbf{m}_k^q \leftarrow \mathrm{Mask}(\mathbf{a}_k^q)$ ▷ $\mathbf{m}_k^q \in \{0,1\}^D$
  end for
end for
// Time courses
for $n = 1, \ldots, N$ **do**
  for $q \in \{\mathrm{A}, \mathrm{B}\}$ **do**
    for $k = 1, \ldots, K$ **do**
      $\mathbf{s}_k^{q,(n)} \leftarrow \mathrm{MeanPool}(\mathbf{X}^{q,(n)}, \mathbf{m}_k^q)$; resample to length L_{c}
    end for
  end for
end for
// Pairwise consistency
for $i = 1, \ldots, K$ **do**
  for $j = 1, \ldots, K$ **do**
    $\rho_{ij}^{(n)} \leftarrow \mathrm{Spearman}(\mathbf{s}_i^{\mathrm{A},(n)}, \mathbf{s}_j^{\mathrm{B},(n)}), \quad \forall n$
    $\bar{\rho}_{ij} \leftarrow \frac{1}{N} \sum_{n=1}^{N} \rho_{ij}^{(n)}$
  end for
end for
$\mathrm{FNF} \leftarrow \max_{i,j} \bar{\rho}_{ij}$
$C \leftarrow \#\{(i,j) : \bar{\rho}_{ij} > \tau_a\}$

Score and aligned-network statistic. The FNF score is the maximum of the consistency matrix,

$$\mathrm{FNF}(\mathcal{M}^{\mathrm{A}}, \mathcal{M}^{\mathrm{B}}) = \max_{i,j} \bar{\rho}_{ij}, \tag{8}$$

and we additionally report the *aligned-network count*, $C = \#\{(i,j) : \bar{\rho}_{ij} > \tau_a\}$ with $\tau_a = 0.4$. The maximum is sensitive to the multiple-comparison issue inherent in taking the best of K^2 entries; we address this directly and quantitatively in Section 6.3, where we show that (i) a null distribution makes the maximum interpretable, and (ii) the aligned-network count, which is a systematic (not extremal) statistic, exhibits a clean dichotomy between homologous and independent model pairs. The full procedure is summarized in Algorithm 1.

6 Experiments

6.1 Setup

We use WikiText-2 Merity et al. (2016) samples (typically 200–300 tokens, truncated to at most 256) so that each time course has enough points for a stable correlation estimate. For each model pair, we feed the *same* input texts to both models. We use $N = 10$ samples to fit CanICA and $N = 100$ samples to estimate time courses, with $K = 64$ components. We compare FNF against REEF Zhang et al. (2025), the strongest representation-based fingerprint able to handle mismatched feature dimensions; we also summarize the qualitative capability of parameter- (PCS/ICS), REEF- and attention-difference-based methods in Table 8.

6.2 Can FNF identify lineage?

Fine-tuning. Table 1 compares each pretrained model with its instruction-tuned counterpart. All three methods give high scores here; these are same-tokenizer pairs, which is where AttnDiff is

Table 1: FNF, REEF and AttnDiff between a pretrained model and its instruction-tuned version. FNF is the largest entry of $\bar{\mathbf{R}}$.

	LLaMA2-7B → LLaMA2-7B-chat	Qwen2.5-7B → Qwen2.5-7B-Instruct
REEF	0.9877	0.9225
AttnDiff	0.9988	0.9913
FNF (ours)	0.9351	0.9196

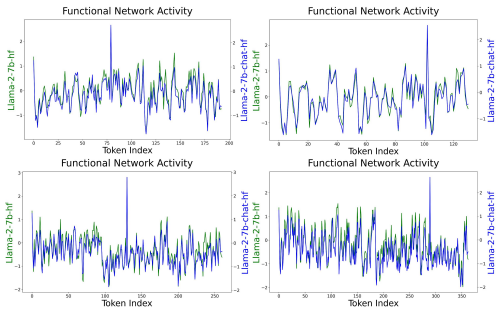


Figure 2: Functional network activity for LLaMA2-7B and LLaMA2-7B-chat across samples, for the most-aligned network pair. The two trajectories co-vary closely, illustrating the high functional-activity consistency FNF detects in derived models.

strongest (Appendix E). FNF preserves the lineage even though fine-tuning changes weights substantially. Figure 2 shows this directly for LLaMA2-7B and LLaMA2-7B-chat: the most-aligned functional network's activity trajectories closely co-vary across samples, even though the networks are extracted independently in each model.

Fine-tuning magnitude. Table 2 scales the fine-tuning budget from 370M to 700B tokens. FNF stays in the strong-correlation regime throughout (0.65–0.95) and does not decay systematically with the budget—the two most heavily fine-tuned models score 0.87 and 0.86—so the fingerprint survives large-scale adaptation; AttnDiff stays near 0.99 throughout, the signature of a score driven by tokenizer sharing rather than lineage (Appendix E). Figure 3 shows the corresponding functional-network activity: even for the model fine-tuned on 700B tokens, the most-aligned network trajectories still track one another closely.

Independent training-data variants. Table 5 (appendix) shows a hard negative: Amber and OpenLLaMA share LLaMA2's *architecture* but were trained from scratch on different data Geng & Liu (2023); Liu et al. (2023c). FNF correctly yields low scores (0.3142 and 0.3757, versus 0.3673 and 0.4159 for REEF), showing it detects lineage rather than architecture or tokenizer. AttnDiff shows the opposite failure mode on the same two models: it scores LLaMA2-7B/Amber at 0.9687, above three of the same-family pairs in Table 4, yet scores LLaMA2-7B/OpenLLaMA at only 0.2453, solely because Amber uses the LLaMA-2 tokenizer and OpenLLaMA the LLaMA-1 one (Appendix E). Figure 4 confirms this at the level of the activity itself: even the most correlated network pair of these independent models fails to align.

Cross-model comparison. Table 6 (appendix) compares across families and within the Qwen/GLM families. REEF reports high similarity even for clearly unrelated models (Qwen2.5-7B vs. GLM4-9B: 0.8567), whereas FNF gives low cross-family scores while retaining moderate-to-strong in-family scores.

Weight repackaging. Table 3 is our key new-attack scenario. Mistral-7B → Solar-10.7B (depth 32→48) and Mistral-11B are depth-expanded models that inherit Mistral weights Kim et al. (2024b); Jiang et al. (2023). FNF identifies their shared origin; REEF fails on both repackaged pairs—Mistral-7B vs. Solar-10.7B scores 0.32 and Solar-10.7B vs. Mistral-11B 0.33, below the median unrelated pair—and is high only for Mistral-7B vs. Mistral-11B (0.99). AttnDiff also flags all three, but it scores an unrelated pair that merely shares a tokenizer nearly as highly (0.9687 against 0.9850), so its score is only weak evidence of repackaging specifically (Appendix E).

Table 2: FNF, REEF and AttnDiff between LLaMA2-7B-chat and LLaMA2 fine-tuned on increasing amounts of data.

Tokens	Vicuna-v1.5 370M	Chinesellama-2-7B 13B	Codellama-7B 500B	Llemma-7B 700B
REEF	0.7396	0.9983	0.8833	0.9701
AttnDiff	0.9989	0.9986	0.9915	0.9854
FNF (ours)	0.6463	0.9454	0.8667	0.8593

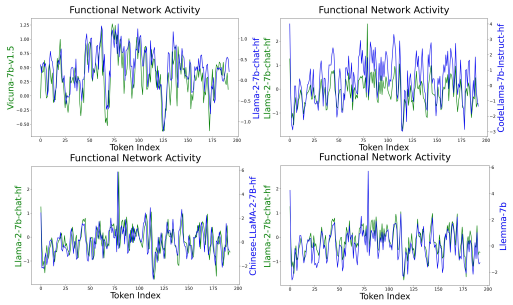


Figure 3: Functional network activity of LLaMA2-7B-chat and its fine-tuned variants trained on different amounts of data. The most-aligned network pair still co-varies after 700B tokens of fine-tuning.

Other disguise attacks. In Appendix A, we report pruning (structured width/depth and unstructured, 50% removal), model merging, parameter permutation, and scaling transformation: FNF preserves a strong signature while REEF's margin degrades (e.g., REEF drops to $\sim$0.001–0.28 for Wanda-sp pruned models). Table 8 (appendix) gives a qualitative capability summary.

6.3 Interpreting and calibrating the score

We use FNF as a calibrated verification score rather than a measure of lineage distance: it reports the strongest functional alignment between independently extracted networks, and a decision rule is obtained by calibrating it on labeled pairs. Because the score is a *maximum* over $K \times K$ network pairs, two questions arise: whether a high value could occur by chance, and how to use the score for verification. We address them with a labeled evaluation set, an empirical null analysis, and the aligned-network count.

We assemble a labeled set of model pairs and organize it by provenance: *verified derivation* (documented weight inheritance via instruction tuning, fine-tuning, continued pretraining, pruning, merging, or depth repackaging; $n = 8$), *same-family* (models from the same family but trained independently—different Qwen sizes, or independently trained descendants of a common ancestor; $n = 11$), and *independent* (different lineage, or the same architecture trained from scratch; $n = 21$); the full set is listed in Appendix D. Because REEF and FNF use different score scales, we evaluate both as classifiers using ROC-AUC. As shown in Table 4, FNF separates verified derivation from independent pairs perfectly (AUC 1.00; bootstrap 95% CI over pairs $[1.00, 1.00]$) and likewise separates same-family from independent pairs. Recomputing REEF with its released protocol on the same 40 pairs gives AUCs of 0.89 and 0.81—well below FNF—and the repackaged Mistral–Solar pair scores 0.32, below the median unrelated pair. AttnDiff Zhang et al. (2026) reaches AUC 1.00 on verified derivation but only 0.97 and 0.98 on the other two comparisons, its residual errors being unrelated pairs that happen to share a tokenizer, which it scores nearly as high as the repackaged pairs (Appendix E). Because some models occur in several pairs, the pair-level bootstrap in Table 4 is descriptive rather than the final estimate. We therefore repeat the analysis with the sampling unit moved to the 20 underlying models, to whole families, and by leaving out one model at a time. FNF stays at 1.00 under all three schemes; REEF's interval widens to $[0.47, 1.00]$ once models rather than pairs are resampled. Holding out one model family at a time and calibrating on the rest likewise yields a held-out AUC of 1.00. The gap is widest at the operating point a deployment would use: at zero false positives FNF recovers every related pair (any threshold in $[0.38, 0.40]$), whereas REEF recovers 37% and AttnDiff, on the 36 pairs it can score, 83%; REEF needs an FPR of 0.30 to reach a true-positive rate of 0.89.

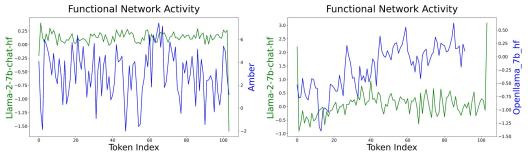


Figure 4: Functional network activity of the independently trained Amber and OpenLLaMA-7B, which show little alignment with LLaMA2. Compare with the derived-model pairs in Figures 2 and 3.

Table 3: Weight-repackaging detection. FNF identifies common origin where REEF fails; AttnDiff is high for all three pairs simply because all three share the Mistral tokenizer (Appendix E).

Pair	REEF	AttnDiff	FNF (ours)
Mistral-7B & Solar-10.7B	0.3226	0.9850	0.5743
Mistral-7B & Mistral-11B	0.9923	0.9989	0.5820
Solar-10.7B & Mistral-11B	0.3320	0.9868	0.7958

The pairs above are labeled from model cards and release notes, so a single mislabelled pair would invalidate the AUC. The Pythia suite Biderman et al. (2023) offers provenance that can be *verified* rather than documented: all of its models come from one training run released at 154 checkpoints, so two checkpoints of the same model are a genuine case of continued training, while models of different sizes were initialized and trained separately on the same data and recipe. We therefore build a second benchmark from it—15 derivation pairs (six checkpoints of Pythia-410M, steps $1{,}000$–$143{,}000$), 15 same-family pairs (six sizes from 70M to 2.8B) and 12 independent pairs (those sizes against Qwen2.5-3B and Vicuna)—and repeat the evaluation without relying on any documentation. Related pairs again lie entirely above independent ones (AUC 1.00 for both), whereas the same REEF protocol reaches only 0.69 and 0.41, i.e. at or below chance for the same-family comparison.

This benchmark also exposes an informative failure mode. The one comparison that is not perfect is derivation vs. same-family (AUC 0.92), and every one of the 18 inverted pairs involves the step-$1{,}000$ checkpoint, which has seen 0.7% of training and whose functional organization has not yet formed; it scores 0.38–0.54, inside the same-family range (0.29–0.51). Excluding that checkpoint the classes separate completely (0.84–0.97 against at most 0.51). A model whose functional networks are not yet stable cannot be expected to match its own descendants.

Because absolute scores also depend on model scale, pooling the two benchmarks into one threshold is not meaningful—the pooled same-family-vs. independent AUC is 0.98—so we calibrate within a benchmark of matched scale (Table 4).

The maximum over $K^2 = 4096$ pairs raises a multiple-comparison concern: could an unrelated pair score high simply because so many pairs are tested? Because the trajectories are strongly autocorrelated and the ICA components are estimated from the same data, an i.i.d. null would be far too optimistic, so we estimate the null of the *entire maximum statistic* empirically, destroying the temporal correspondence in three ways (circular shift, block permutation, time shuffle) over 100 permutations each. As Table 16 reports, the chance level is 0.03–0.08 and, as Appendix B shows, it stays there across model scales from 70M to 7B, four architectures and K from 16 to 64; a related pair (Qwen2.5-7B vs. Qwen2.5-3B) reaches 0.60, an order of magnitude above it. The null thus rules out a multiple-comparison explanation, but it is deliberately not our decision rule: two unrelated models reading the same texts still share a stimulus-driven component (0.23). Verification uses the calibrated operating point instead.

The maximum alone does not show whether the alignment is concentrated in one lucky network pair, so we also report the *aligned-network count*, $C = \#\{(i,j) : \bar{\rho}_{ij} > \tau_a\}$, which measures how broadly the consistency is distributed. As Table 7 shows, homologous pairs exhibit 5–44 aligned network pairs whereas none of the six independent pairs exhibits even one: the signal is not confined to a single unusually correlated pair.

Two controls characterize the score. First, specificity—replacing the functional networks with random neuron groups of the same sizes lowers a related pair's score from 0.60 to 0.39, and the correlation between the two models' whole-layer mean trajectories is only 0.06, so the score is neither

Table 4: Calibrated evaluation on the 7B benchmark (8 derivation / 11 same-family / 21 independent pairs). Bootstrap 95% CIs over pairs are shown in brackets; REEF is recomputed with the released protocol on all 40 pairs (Appendix D). FNF separates the classes perfectly in every comparison; REEF does not, and its shortfall is concentrated at the low false-positive rates a deployment would use. †AttnDiff needs per-head attention maps, which the ChatGLM models do not expose, so its column is computed on the 36 pairs it supports (8/10/18); on those same pairs FNF scores 1.00/1.00/1.00 and REEF 0.92/0.89/0.90 (Appendix E).

Comparison	n_+/n_-	FNF	REEF	AttnDiff†
Verified derivation vs. independent	8/21	**1.00** [1.00,1.00]	0.89 [0.70,1.00]	1.00 [1.00,1.00]
Same-family vs. independent	11/21	**1.00** [1.00,1.00]	0.81 [0.65,0.95]	0.97 [0.89,1.00]
Related (both) vs. independent	19/21	**1.00**	0.84 [0.71,0.95]	0.98 [0.94,1.00]



Figure 5: Cumulative mean FNF as a function of the number of samples, for a derived pair (ChatGLM2–ChatGLM3, left) and an independent pair (ChatGLM2–LLaMA2-7B-chat, right). A few dozen samples suffice to separate the two regimes.

generic activation similarity nor an artifact of the layer mean. Second, stability—repeating CanICA with different random seeds varies the score by ± 0.05, comparing a model with *itself* across seeds yields 0.98 (well above the 0.60 of a related pair), and varying K between 32 and 128 moves it between 0.50 and 0.71 (Appendix B). The score is thus stable in the network *dynamics* even though the components themselves are not uniquely identified (best-match IoU across seeds 0.33, Table 20).

Individual scores therefore need not be ordered by lineage proximity: Mistral-7B vs. Mistral-11B (0.5820) scores below Solar-10.7B vs. Mistral-11B (0.7958), because CanICA is run independently for each model and ICA components have no canonical ordering.

Finally, Appendix A examines K and the input dataset: given long enough sequences the score does not depend on the text distribution. It also stabilizes within tens of samples, which Figure 5 shows for the homologous pair ChatGLM2-6B/ChatGLM3-6B and the independent pair ChatGLM2-6B/LLaMA2-7B-chat.

7 Discussion and Conclusion

Inspired by functional brain networks in neuroimaging, we proposed FNF, a white-box fingerprint for LLM lineage verification that requires no fingerprint-specific training. Lineage information can persist in a model's *functional organization*: neurons that participate in the same functional networks keep consistent activation dynamics under the same inputs, even after fine-tuning, pruning, parameter permutation, merging, or weight repackaging. Comparing these dynamics rather than individual parameters or representation dimensions lets FNF operate across models with different architectures, hidden dimensions, and tokenizers.

In practice a verifier feeds the same texts to both models, extracts their functional networks, and compares the resulting activity time courses; the aligned-network count then shows how broadly the consistency is distributed. The signal survives every transformation we tested and, unlike weight- and representation-based fingerprints, requires no architectural or dimensional correspondence.

FNF also has limitations: as a white-box method it requires access to internal activations and does not support API-only verification, and its ICA decomposition may not capture all of the structure of modern representations, so nonlinear or learned decompositions are worth exploring. Broader evaluation across more families and scales would further characterize the generality of the signal. We view FNF as a practical and interpretable approach to LLM provenance verification, and as evidence that functional organization is a persistent source of fingerprinting signal.

AI USE STATEMENT

Generative AI tools were used to assist with experimental code and the drafting of parts of this manuscript. The authors reviewed all AI-assisted output, verified the experimental code and results, and take full responsibility for the final content.

REFERENCES

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX security symposium (USENIX Security 18)*, pp. 1615–1631, 2018.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198, 2009.

Jialuo Chen, Jingyi Wang, Tinglan Peng, Youcheng Sun, Peng Cheng, Shouling Ji, Xingjun Ma, Bo Li, and Dawn Song. Copy, right? a testing framework for copyright protection of deep learning models. In *2022 IEEE symposium on security and privacy (SP)*, pp. 824–841. IEEE, 2022.

Jialuo Chen, Youcheng Sun, Jingyi Wang, Peng Cheng, and Xingjun Ma. Deepjudge: a testing framework for copyright protection of deep learning models. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pp. 64–67. IEEE, 2023.

Emily S Finn, Xilin Shen, Dustin Scheinost, Monica D Rosenberg, Jessica Huang, Marvin M Chun, Xenophon Papademetris, and R Todd Constable. Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. *Nature neuroscience*, 18(11):1664–1671, 2015.

Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, May 2023. URL `https://github.com/openlm-research/open_llama`.

Jia Guo and Miodrag Potkonjak. Watermarking deep neural networks for embedded systems. In *2018 IEEE/ACM international conference on computer-aided design (ICCAD)*, pp. 1–8. IEEE, 2018.

Mengshen He, Xiangyu Hou, Enjie Ge, Zhenwei Wang, Zili Kang, Ning Qiang, Xin Zhang, and Bao Ge. Multi-head attention-based masked sequence model for mapping functional brain networks. *Frontiers in Neuroscience*, 17:1183145, 2023.

Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

Dmitri Iourovitski, Sanat Sharma, and Rakshak Talwar. Hide and seek: Fingerprinting large language models with evolutionary learning. *arXiv preprint arXiv:2408.02871*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, and Hyoung-Kyu Song. Shortened llama: Depth pruning for large language models with comparison of retraining methods. *arXiv preprint arXiv:2402.02834*, 2024a.

Sanghoon Kim, Dahyun Kim, Chanjun Park, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, et al. Solar 10.7 b: Scaling large language models with simple yet effective depth up-scaling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pp. 23–35, 2024b.

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, pp. 14991–14999, 2023.

Zheng Li, Chengyu Hu, Yang Zhang, and Shanqing Guo. How to prove your model belongs to you: A blind-watermark based framework to protect intellectual property of dnn. In *Proceedings of the 35th annual computer security applications conference*, pp. 126–137, 2019.

Yiheng Liu, Enjie Ge, Ning Qiang, Tianming Liu, and Bao Ge. Spatial-temporal convolutional attention for mapping functional brain networks. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–4. IEEE, 2023a.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, pp. 100017, 2023b.

Yiheng Liu, Enjie Ge, Mengshen He, Zhengliang Liu, Shijie Zhao, Xintao Hu, Ning Qiang, Dajiang Zhu, Tianming Liu, and Bao Ge. Mapping dynamic spatial patterns of brain function with spatial-wise attention. *Journal of Neural Engineering*, 21(2):026005, 2024a.

Yiheng Liu, Enjie Ge, Zili Kang, Ning Qiang, Tianming Liu, and Bao Ge. Spatial-temporal convolutional attention for discovering and characterizing functional brain networks in task fmri. *NeuroImage*, 287:120519, 2024b.

Yiheng Liu, Xiaohui Gao, Haiyang Sun, Bao Ge, Tianming Liu, Junwei Han, and Xintao Hu. Brain-inspired exploration of functional networks and key neurons in large language models. *arXiv preprint arXiv:2502.20408*, 2025a.

Yiheng Liu, Hao He, Tianle Han, Xu Zhang, Mengyuan Liu, Jiaming Tian, Yutong Zhang, Jiaqi Wang, Xiaohui Gao, Tianyang Zhong, Yi Pan, Shaochen Xu, Zihao Wu, Zhengliang Liu, Xin Zhang, Shu Zhang, Xintao Hu, Tuo Zhang, Ning Qiang, Tianming Liu, and Bao Ge. Understanding llms: A comprehensive overview from training to inference. *Neurocomputing*, 620:129190, 2025b.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*, 2023c.

Hope McGovern, Rickard Stureborg, Yoshi Suhara, and Dimitris Alikaniotis. Your large language models are leaving fingerprints. *arXiv preprint arXiv:2405.14057*, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. pp. 24950–24962, 2023.

Dario Pasquini, Evgenios M Kornaropoulos, and Giuseppe Ateniese. {LLMmap}: Fingerprinting for large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 299–318, 2025.

Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Benjamin Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. Are you copying my model? protecting the copyright of large language models for eaas via backdoor watermark. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

Ning Qiang, Qinglin Dong, Hongtao Liang, Jin Li, Shu Zhang, Cheng Zhang, Bao Ge, Yifei Sun, Jie Gao, Tianming Liu, et al. Learning brain representation using recurrent wasserstein generative adversarial net. *Computer Methods and Programs in Biomedicine*, 223:106979, 2022.

Ning Qiang, Qinglin Dong, Heng Huang, Han Wang, Shijie Zhao, Xintao Hu, Qing Li, Wei Zhang, Yiheng Liu, Mengshen He, et al. Deep learning in functional brain mapping and associated applications. In *Deep Learning for Medical Image Analysis*, pp. 395–423. Elsevier, 2024.

Marcus E Raichle. The brain's default mode network. *Annual review of neuroscience*, 38(1):433–447, 2015.

Mark Russinovich and Ahmed Salem. Hey, that's my model! introducing chain & hash, an llm fingerprinting technique. *arXiv preprint arXiv:2407.10887*, 2024.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.

Stephen M Smith, Peter T Fox, Karla L Miller, David C Glahn, P Mickle Fox, Clare E Mackay, Nicola Filippini, Kate E Watkins, Roberto Toro, Angela R Laird, et al. Correspondence of the brain's functional architecture during activation and rest. *Proceedings of the national academy of sciences*, 106(31):13040–13045, 2009.

Dimitri Van De Ville, Younes Farouj, Maria Giulia Preti, Raphaël Liégeois, and Enrico Amico. When makes you unique: Temporality of the human brain fingerprint. *Science advances*, 7(42): eabj0751, 2021.

Gaël Varoquaux, Merlin Keller, Jean-Baptiste Poline, Philippe Ciuciu, and Bertrand Thirion. Ica-based sparse features recovery from fmri datasets. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1177–1180. IEEE, 2010a.

Gaël Varoquaux, Sepideh Sadaghiani, Philippe Pinel, Andreas Kleinschmidt, Jean-Baptiste Poline, and Bertrand Thirion. A group model for stable multi-subject ica on fmri datasets. *Neuroimage*, 51(1):288–299, 2010b.

Jiaqi Wang, Enze Shi, Huawen Hu, Chong Ma, Yiheng Liu, Xuhui Wang, Yincheng Yao, Xuan Liu, Bao Ge, and Shu Zhang. Large language models for robotics: Opportunities, challenges, and perspectives. *Journal of Automation and Intelligence*, 2024.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. Llmdet: A third party large language models generated text detection tool. *arXiv preprint arXiv:2305.15004*, 2023.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3277–3306, 2024.

Zhenhua Xu, Xubin Yue, Zhebo Wang, Qichen Liu, Xixiang Zhao, Jingxuan Zhang, Wenjun Zeng, Wengpeng Xing, Dezhang Kong, Changting Lin, et al. Copyright protection for large language models: A survey of methods, challenges, and trends. *arXiv preprint arXiv:2508.11548*, 2025.

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. *Advances in Neural Information Processing Systems*, 37:126332–126362, 2024.

Haobo Zhang, Zhenhua Xu, Junxian Li, Shangfeng Sheng, Dezhang Kong, and Meng Han. Attndiff: Attention-based differential fingerprinting for large language models. *arXiv preprint arXiv:2604.05502*, 2026.

Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia conference on computer and communications security*, pp. 159–172, 2018.

Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. REEF: Representation encoding fingerprints for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SnDmPkOJ0T.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

A SUPPLEMENTARY: ADDITIONAL RESULTS

Implementation details. Hidden states are collected with `output_hidden_states`, the embedding output is dropped, and the L block outputs are stacked along the feature axis, so $D = L \cdot h$ (e.g. 32×4096 for a 7B LLaMA). Inputs are truncated to 256 tokens, and the D channels are z-scored across tokens before decomposition. CanICA takes the top-K left singular vectors of $\mathbf{X}^\top$ and runs FastICA with the cube nonlinearity ($\text{tol} = 10^{-4}$, at most 300 iterations, seed 666); the resulting maps are z-scored and thresholded at $|z| > 3.6$, which retains under 2% of channels. Time courses are averaged over masked channels using the raw, un-z-scored activations, and alignment uniformly downsamples the longer time course onto the shorter length. Models are loaded in `float16` and activations are cast to `float32` for the decomposition; extracting a 7B signature (100 samples) takes a few minutes on four consumer GPUs, after which scoring a pair is a single 64×64 Spearman computation.

Training-data variants. Table 5 reports all three methods between LLaMA2-7B and independently trained models that share its architecture.

Table 5: FNF, REEF and AttnDiff between LLaMA2-7B and independently trained models with identical architecture (Amber and OpenLLaMA-7B share LLaMA2's architecture but not its tokenizer: OpenLLaMA uses the LLaMA-1 one).

	Amber	OpenLLaMA-7B
REEF	0.3673	0.4159
AttnDiff	0.9687	0.2453
FNF (ours)	0.3142	0.3757

Cross-model comparison. Table 6 reports FNF and REEF across families and within the Qwen/GLM families.

Table 6: FNF/ REEF across model families and within families. FNF separates in-family (derived) from cross-family (independent) pairs much more sharply than REEF.

	Qwen2.5-7B-Instruct vs.		
	Qwen1.5-7B-IT	Qwen2.5-3B-IT	Qwen2.5-1.5B-IT
REEF	0.7958	0.7988	0.6832
FNF (ours)	0.4043	0.6017	0.4705
	LLaMA3-8B	ChatGLM3-6B	GLM4-9B
REEF	0.7944	0.6842	0.8567
FNF (ours)	0.2586	0.1507	0.2737
	ChatGLM2-6B vs. LLaMA3-8B	ChatGLM3-6B	GLM4-9B
REEF	0.9071	0.7475	0.8845
FNF (ours)	0.2612	0.5882	0.3626

Aligned-network count. Table 7 lists, for each model pair, how many of the $K \times K$ network pairs have mean consistency $\bar{\rho} > 0.4$.

Capability comparison. Table 8 summarizes which fingerprinting methods handle each model-disguise scenario.

Dataset robustness. Table 9 reports the effect of the input dataset. FNF is robust to the choice of text as long as the sequences are long enough: WikiText-2, SQuAD and random token sequences all give scores in the 0.74–0.87 range, whereas SST-2 drops to 0.44 because its samples are only about 10 tokens long. The random-token result is the informative control here: it shows that FNF does not require semantic content—a shared stimulus is sufficient to elicit the inherited functional dynamics.

Table 7: Aligned-network count (pairs with $\bar{\rho} > 0.4$) between model pairs. A strong, consistent dichotomy.

Model pair	Homology	Aligned nets
ChatGLM2-6B / ChatGLM3-6B	Homologous	5
LLaMA2-7B / LLaMA2-7B-chat	Homologous	19
LLaMA2-7B-chat / Llemma-7B	Homologous	11
LLaMA2-7B / LLaMA2-7B-shuffled	Homologous	31
ChatGLM3-6B / ChatGLM3-6B-shuffled	Homologous	38
Evollm-jp-7B / Shisa-gamma-7B	Homologous	44
Solar-10.7B / Mistral-11B	Homologous	27
LLaMA2-7B / ChatGLM2-6B	Independent	0
Qwen2.5-7B-IT / LLaMA3-8B	Independent	0
Qwen2.5-7B-IT / ChatGLM3-6B	Independent	0
Qwen2.5-7B-IT / GLM4-9B	Independent	0
LLaMA2-7B / Amber	Independent	0
LLaMA2-7B / OpenLLaMA-7B	Independent	0

Table 8: Capability of fingerprinting methods across model-disguise scenarios. A checkmark means that the published protocol can be applied *directly* in the deployment setting we study—the suspect model is available as weights, no architectural, layer or feature-dimension correspondence may be assumed, and retraining is not possible; it is not a claim about accuracy in settings we did not test. A cross means either that the protocol cannot be applied in this setting at all (PCS and ICS, which assume a parameter or a dimensional correspondence) or that it applies but fails to separate the scenario from unrelated models on the pairs we evaluated: REEF scores the structured-width-pruned models and the two repackaged pairs below the median unrelated pair and rates an unrelated cross-family pair above every in-family Qwen pair, while AttnDiff separates the three repackaged pairs from every independent pair in our set but cannot separate unrelated cross-architecture models from related ones, its score being dominated by shared tokenization (Appendix E). PCS/ICS = parameter-based; REEF = representation-based; FNF = ours.

Scenario	PCS	ICS	REEF	AttnDiff	FNF (Ours)
Fine-tuning	×	×	✓	✓	✓
Same-architecture training variants	✓	✓	✓	✓	✓
Parameter permutation	×	×	✓	✓	✓
Pruning (depth, unstructured)	×	×	✓	✓	✓
Structured width pruning	×	×	×	✓	✓
Model merging	✓	✓	✓	✓	✓
Scaling transformation	✓	✓	✓	✓	✓
Cross-model comparison	×	×	×	×	✓
Weight repackaging	×	×	×	✓	✓

Table 9: FNF for LLaMA2-7B-chat vs. Llemma-7B across datasets. Random-token inputs are gibberish (no semantics).

Dataset	WikiText-2	SQuAD	SST-2	Random tokens
FNF	0.8718	0.8514	0.4353	0.7426

Component count K. Table 10 shows that the separation between related and independent pairs persists across a broad range of K, although the absolute score varies with K; this is why we fix $K = 64$ and calibrate a decision threshold rather than compare raw scores across settings.

Table 10: FNF under different K; the $K = 64$ column is the setting used throughout the paper.

K	10	20	40	64	128
ChatGLM2-6B vs. ChatGLM3-6B	0.3201	0.3844	0.6202	0.5882	0.5101
Qwen2.5-7B-IT vs. LLaMA3-8B	0.1970	0.3069	0.3042	0.2586	0.3442

Pruning, merging, permutation, scaling. Table 11, Table 12, Table 13, Table 14 extend the robustness analysis to pruning (structured width/depth and unstructured, 50% removal), model merging, parameter permutation, and scaling transformation. In each tested setting FNF preserves a strong signature of the original model; REEF matches it on depth and unstructured pruning but degrades markedly under structured width removal.

Table 11: Pruning robustness (50% removal) for LLaMA2-7B-chat and Vicuna-v1.5-7B. FNF remains strong.

	FLAP (width)	Wanda (unstruct.)	Shortened (depth)	Wanda-sp (width)
LLaMA2 (REEF)	0.8236	0.9991	0.9996	0.2767
LLaMA2 (FNF)	0.6172	0.9451	0.5965	0.5007
Vicuna (REEF)	0.9863	0.9107	0.9100	0.0012
Vicuna (FNF)	0.9097	0.9249	0.9692	0.6751

Spatial overlap. Table 15 relates the spatial similarity (IoU) of functional networks to their FNF score. For homologous models high IoU corresponds to high FNF; for independent models this breaks down (the pair with max IoU does not yield max FNF, and the max-FNF pair has IoU zero), suggesting that FNF is not reducible to spatial-map overlap alone.

B Additional statistical analyses

Empirical null of the maximum statistic. Table 16 reports, for a related pair (Qwen2.5-7B-Instruct vs. Qwen2.5-3B-Instruct) and an independent pair (Qwen2.5-7B-Instruct vs. Vicuna-v1.5), the observed FNF score together with the null distribution of $\max_{i,j} \bar{\rho}_{ij}$ obtained by destroying the temporal correspondence between the two models' trajectories in three different ways (100 permutations each). Table 17 repeats the analysis over ten pairs spanning model scales from 70M to 7B parameters and four architectures, with 200 permutations per cell, and reports the permutation p-value of each observed score.

The null level is essentially universal. Across the ten pairs of Table 17—70M to 7B parameters, GPT-NeoX, Qwen2.5, LLaMA-2 and Mistral—the 95th percentile of the null stays between 0.033 and 0.081, with no trend in model scale or architecture, and it grows only mildly with the number of components (Table 18) while the observed score grows much faster ($0.35 \to 0.60$ and $0.24 \to 0.91$ as K goes from 16 to 64). Related pairs sit 6–11$\times$ above this floor, and independently trained pairs 3–5$\times$ above it, because two unrelated models that read the same texts still share a small stimulus-driven component. Every observed score therefore has $p \leq 0.005$ against the temporal null: that null bounds the multiple-comparison artifact of taking a maximum over K^2 pairs, but it is not by itself a lineage test. The decision rule is calibrated instead against the distribution of scores of independently trained models. Within the 7B benchmark the largest of the 21 independent scores is 0.376 while all 19 derivation and same-family pairs lie above it, so any threshold in $[0.38, 0.40]$ gives zero false positives at full recall there; we use $\tau = 0.42$, which still has zero false positives on the whole

Table 12: Model-merging robustness: Evollm-jp-7B and its merged components.

	Shisa-gamma-7B	WizardMath-7B	Abel-7B-002
REEF	0.9635	0.9526	0.9374
FNF	0.9595	0.8442	0.9589

Table 13: Parameter-permutation robustness.

	LLaMA2-7B-chat	Vicuna-v1.5-7B	ChatGLM3-6B
REEF	1.0000	1.0000	1.0000
FNF	0.9986	0.9972	0.9971

benchmark and detects all eight verified descendants, at the cost of the single lowest-scoring same-family pair (0.40); across the smaller Pythia models the two distributions are cleanly separated but shifted (0.29–0.97 for related pairs, 0.15–0.27 for independent ones), which is why the threshold is best calibrated for the model scale in use rather than transferred as an absolute constant.

Comparison with the parametric Student-t approximation. A standard shortcut would treat each of the K^2 network-pair correlations as an independent draw with a Student-t null, which puts the 95th percentile of the maximum at $\Phi^{-1}(0.95^{1/K^2})/\sqrt{(T-1)N}$ for trajectories of T tokens and N samples. Table 19 evaluates that approximation on the same ten pairs. It is anti-conservative by a factor of about 1.5 on average (0.030–0.045 against an empirical 0.041–0.081; on one pair it is even mildly conservative), because the trajectories are temporally autocorrelated and the K^2 comparisons are positively dependent. Correcting the calculation for the measured lag-1 autocorrelation of the trajectories (mean $\bar{\rho}_1 = 0.27$, so that the effective length is $T_{\text{eff}} = T(1-\bar{\rho}_1)/(1+\bar{\rho}_1) \approx 100$ rather than ≈ 180) recovers most of the difference (0.041–0.060, a factor of 1.16). We therefore report the empirical permutation level, which is the conservative of the three.

ICA stability and controls. Table 20 reports the sensitivity of the score to K and to the CanICA random seed, the score obtained when the ICA networks are replaced by random neuron groups of identical sizes, and the correlation of the two models' whole-layer mean trajectories.

Alignment ablation. Table 21 varies how the two models' trajectories are aligned. Aligning by character offset—a tokenizer-independent *text* alignment—raises the score of an unrelated pair from 0.23 to 0.35, because text-aligned activations share a strong stimulus-driven component. The default uniform alignment does not require token correspondence and therefore suppresses this confound while keeping a related pair high (0.60). Within a shared tokenizer, where the default alignment already aligns text, FNF still separates derived from independent models (Table 2, Table 5). Controls that destroy the temporal order—reversing or shuffling the time axis—collapse the score to 0.03–0.19. Temporal alignment is therefore an explicit design choice rather than an implementation detail.

C A LARGER BENCHMARK WITH EXACTLY KNOWN PROVENANCE

The Pythia suite Biderman et al. (2023) offers provenance that is exact rather than documented: its models are released at 154 training steps of a single run, so two checkpoints of the same model are a genuine instance of continued-training derivation, while models of different sizes are separately initialized and trained—sharing the architecture family, tokenizer, and training data, but not weights. We therefore use it to build a larger benchmark: 15 derivation pairs (all pairs of six checkpoints of Pythia-410M, spanning steps 1,000–143,000), 15 same-family pairs (all pairs of six Pythia sizes from 70M to 2.8B), and 12 independent pairs (the six sizes against Qwen2.5-3B-Instruct and Vicuna-7B, under identical stimuli). Table 22 summarizes the outcome. The three categories are cleanly ordered and non-overlapping between related and independent pairs, and the AUCs are 1.00 (derivation vs. independent), 1.00 (same-family vs. independent), and 0.92 (derivation vs. same-family). Early checkpoints (1,000 steps) sit at the low end of the derivation range, consistent with functional organization still forming; this is why FNF is best read as a functional-consistency score rather than a lineage distance.

Table 14: Scaling-transformation robustness.

	LLaMA2-7B-chat	Qwen2.5-7B-Instruct	ChatGLM3-6B
REEF	1.0000	1.0000	1.0000
FNF	0.9984	0.9832	0.9999

Table 15: Highest spatial similarity (IoU) and the corresponding FNF for LLaMA2-7B-chat vs. fine-tuned and training-data variants.

LLaMA2-7B	Vicuna	Chinesellama	Codellama	Llemma	OpenLLaMA	Amber
IoU	0.6115	0.6161	0.3971	0.3375	0.0735	0.1072
FNF	0.6299	0.9498	0.8405	0.8397	0.3757	0.3142

D CALIBRATED EVALUATION DATA

Table 23 lists the labeled set used for the ROC analysis in Section 6.3, grouped by provenance: *Deriv.* denotes verified derivation (documented weight inheritance), *Fam.* same-family models trained independently within the lineage, and *Indep.* independent lineage, or the same architecture trained from scratch. Table 24 then adds the AttnDiff column for the 36 of these pairs it can score; the four involving ChatGLM2-6B or ChatGLM3-6B are excluded because AttnDiff produces no fingerprint for them. FNF scores are the measurements reported in Section 6 and Appendix A; REEF scores are the corresponding values. REEF is scored from the released feature bank through its own protocol (last token of 10 WikiText-2 stimuli at layer 18 of 32, linear CKA between the resulting $10\times d$ matrices). Model names are abbreviated: *LLaMA2-chat* is LLaMA2-7B-chat, *Chinesellama-2-7B* is Chinese-LLaMA-2-7B, *IT* is the instruction-tuned *Instruct* model of the same size, and *Vicuna-v1.5* is Vicuna-7B. No released Mistral-7B feature file exists, so for the six pairs involving Mistral-7B we replay that protocol ourselves; re-extracting a model that does have a released file (Vicuna-7B) this way reproduces both the stored feature matrix and its REEF score against LLaMA2-chat to within 10^{-4}, which validates the replay.

E COMPARISON WITH AN ATTENTION-DIFFERENCE FINGERPRINT

Table 4 also reports AttnDiff Zhang et al. (2026), a recent non-invasive fingerprint that shares FNF's premise of comparing two models without requiring weight or feature-dimension correspondence. This appendix gives the protocol we used, the per-pair scores, and the analysis behind the two claims made in the main text: that AttnDiff's residual errors are concentrated on unrelated pairs that share a tokenizer, and that the margin by which it separates the repackaged pairs is set by tokenization rather than by shared weights.

Protocol. We ran the authors' released implementation with its default configuration (`--mode diff`, `--cka linear`) and did not tune it. For each of the 30 original/corrupted prompt pairs in the released dataset we obtain the eager attention maps of both prompts, crop them to the shorter common sequence length, and reduce every $(\text{layer}, \text{head})$ to the scalar $\|A^{\text{corr}} - A^{\text{orig}}\|_F$. A model's fingerprint is the resulting $30 \times (LH)$ matrix, and two models are compared by linear CKA over the 30 rows. Because the comparison runs over rows (prompt pairs) rather than columns $(\text{layer}, \text{head})$, AttnDiff accommodates models with different LH, which is what makes it applicable in our setting at all.

Table 16: Empirical null of the FNF maximum statistic (100 permutations). Each null construction destroys the temporal correspondence between the two models' trajectories in a different way.

	Qwen2.5-7B / Qwen2.5-3B (related)			Qwen2.5-7B / Vicuna (independent)		
Null construction	observed	null p95	null max	observed	null p95	null max
circular shift	0.602	0.054	0.063	0.231	0.049	0.063
block permutation	0.602	0.058	0.066	0.231	0.046	0.056
time shuffle	0.602	0.035	0.040	0.231	0.035	0.046

Table 17: Null level of the maximum statistic across model scales and architectures (200 permutations per cell, $K = 64$). *null p95* is the largest 95th percentile of the three temporal controls and p the corresponding conservative permutation p-value. The chance level is near 0.06 for every scale, architecture and relation, so the artifact floor of the maximum statistic is not what separates related from independent pairs.

Pair (architecture)	Relation	FNF	null p95	ratio	p
Pythia-410M, steps 1k / 143k (NeoX)	derivation	0.386	0.062	$6\times$	< 0.005
Qwen2.5-7B / Qwen2.5-3B (Qwen)	related	0.602	0.058	$10\times$	< 0.005
CodeLlama-7B / Llemma-7B (LLaMA-2)	same-family	0.908	0.081	$11\times$	< 0.005
Pythia-160M / Pythia-1.4B (NeoX)	same-family	0.339	0.061	$6\times$	< 0.005
Pythia-70M / Pythia-2.8B (NeoX)	same-family	0.331	0.059	$6\times$	< 0.005
Mistral-7B / Vicuna-7B	independent	0.205	0.043	$5\times$	< 0.005
Qwen2.5-3B / Vicuna-7B	independent	0.197	0.058	$3\times$	< 0.005
Pythia-160M / Qwen2.5-3B	independent	0.239	0.058	$4\times$	< 0.005
Pythia-2.8B / Qwen2.5-3B	independent	0.222	0.058	$4\times$	< 0.005
Qwen2.5-7B / Mistral-7B	independent	0.196	0.054	$4\times$	< 0.005

Table 18: Dependence of the observed score and of the null on the number of components, reported as observed / null p95. Here the $K = 16$ and $K = 32$ columns take the leading K components of the $K = 64$ decomposition rather than refitting CanICA, so that the observed score and the null are computed on the same components; Table 20 reports a refitted sweep. The score grows with K considerably faster than its chance level does.

Pair	$K = 16$	$K = 32$	$K = 64$
Qwen2.5-7B / Qwen2.5-3B (related)	0.350 / 0.045	0.382 / 0.050	0.602 / 0.058
CodeLlama-7B / Llemma-7B (same-family)	0.240 / 0.039	0.810 / 0.072	0.908 / 0.081
Pythia-410M, steps 1k / 143k (derivation)	0.308 / 0.056	0.308 / 0.058	0.386 / 0.062
Mistral-7B / Vicuna-7B (independent)	0.160 / 0.036	0.198 / 0.038	0.205 / 0.043

Table 19: Parametric approximations to the null of the maximum statistic, compared with the empirical permutation level of Table 17. Trajectories are $T \approx 173$–204 tokens long with $N = 50$ or 100 samples; the i.i.d. column is $\Phi^{-1}(0.95^{1/K^2})/\sqrt{(T-1)N}$ and the AR(1) column replaces T by $T(1-\bar{\rho}_1)/(1+\bar{\rho}_1)$ using the measured mean lag-1 autocorrelation $\bar{\rho}_1$ of the two models' trajectories. Ignoring autocorrelation understates the chance level by about a third on average (0.039 against 0.059); we report the empirical level.

Pair	Relation	$\bar{\rho}_1$	T_{eff}	i.i.d. p95	AR(1) p95	empirical p95
Pythia-410M, steps 1k / 143k	derivation	0.18	120	0.045	0.055	0.062
Qwen2.5-7B / Qwen2.5-3B	related	0.29	99	0.031	0.043	0.058
CodeLlama-7B / Llemma-7B	same-family	0.32	105	0.030	0.041	0.081
Pythia-160M / Pythia-1.4B	same-family	0.21	113	0.045	0.056	0.061
Pythia-70M / Pythia-2.8B	same-family	0.23	109	0.045	0.057	0.059
Mistral-7B / Vicuna-7B	independent	0.33	100	0.030	0.042	0.043
Qwen2.5-3B / Vicuna-7B	independent	0.29	100	0.044	0.060	0.058
Pythia-160M / Qwen2.5-3B	independent	0.25	104	0.045	0.059	0.058
Pythia-2.8B / Qwen2.5-3B	independent	0.27	100	0.045	0.060	0.058
Qwen2.5-7B / Mistral-7B	independent	0.32	94	0.031	0.044	0.054

Table 20: ICA stability and controls (Qwen2.5-7B-Instruct vs. Qwen2.5-3B-Instruct unless noted).

Quantity	Value
FNF, $K = 32$ / $K = 64$ / $K = 128$	0.50 / 0.60 / 0.71
FNF across CanICA seeds	0.60 / 0.68 / 0.58 (std 0.04)
FNF(model, itself) across CanICA seeds	0.98
FNF(related pair) with random neuron groups	0.39
Layer-mean trajectory correlation, related / independent	0.06 / -0.04
Best-match spatial-map IoU across CanICA seeds	0.33

Table 21: Alignment ablation on Qwen2.5-7B-Instruct vs. Qwen2.5-3B-Instruct (related) and vs. Vicuna-7B (independent).

Alignment	Related	Independent
uniform, no token correspondence (default)	0.602	0.231
character offset (text aligned)	0.533	0.352
reversed time axis (control)	0.130	0.187
shuffled time axis (control)	0.040	0.029

Table 22: Larger provenance-controlled benchmark from the Pythia suite Biderman et al. (2023) (42 pairs). Derivation pairs are the same run at different steps; same-family pairs are different Pythia sizes (independent initialization, shared data and recipe); independent pairs cross model families. The two AUC columns are computed against the independent pairs, which therefore form the negative class and have no AUC of their own (*ref.*); REEF is evaluated with the same last-token CKA protocol as in Appendix D.

Category	n	FNF range	FNF AUC	REEF AUC
Derivation (same run, different steps)	15	0.38–0.97	**1.00**	0.69
Same-family (different sizes)	15	0.29–0.51	**1.00**	0.41
Independent (different family)	12	0.15–0.27	*ref.*	*ref.*

Coverage. AttnDiff needs per-head attention maps, and the ChatGLM-2 and ChatGLM-3 implementations do not expose them: with `output_attentions=True` and an eager attention implementation, the forward pass still returns no attention tensors for either model. **AttnDiff therefore cannot be applied to ChatGLM-2 or ChatGLM-3 at all**, and its comparison covers 36 of the 40 pairs (8 derivation, 10 same-family, 18 independent). To keep the three methods on identical pairs we recompute FNF and REEF on the same 36 pairs, so the columns in Table 24 differ slightly from those in Table 4; FNF scores 1.00/1.00/1.00 and REEF 0.92/0.89/0.90 (with bootstrap 95% CIs over pairs $[1.00, 1.00]$, $[0.88, 1.00]$, $[0.91, 1.00]$ and $[0.76, 1.00]$, $[0.74, 1.00]$, $[0.79, 0.98]$ respectively).

Where AttnDiff fails. On verified derivation AttnDiff is perfect (AUC 1.000, CI $[1.00, 1.00]$), but on same-family against independent it reaches 0.967 (CI $[0.89, 1.00]$) and on all related pairs 0.981 (CI $[0.94, 1.00]$). The reason is visible in the score ranges: FNF's related and independent scores are disjoint ($[0.4043, 0.9454]$ against $[0.1257, 0.3757]$, a margin of $+0.029$ at the closest point), whereas AttnDiff's overlap ($[0.8427, 0.9989]$ against $[0.1652, 0.9687]$, a margin of -0.126). The independent pair LLaMA2-7B/Amber scores 0.9687—higher than three related pairs, namely Qwen2.5-7B-IT/Qwen2.5-1.5B-IT (0.8427), Qwen2.5-7B-IT/Qwen1.5-7B-IT (0.9061) and Qwen2.5-7B-IT/Qwen2.5-3B-IT (0.9670); Qwen2.5-7B-IT/GLM4-9B (0.9340) is above the two lowest related pairs and Qwen2.5-7B-IT/LLaMA3-8B (0.8739) above the lowest of them. At zero false positives ($\tau \approx 0.97$ for this subset) AttnDiff reaches a true-positive rate of 0.83; at the $\tau = 0.98$ operating point its TPR is 0.78.

Why: the score is dominated by tokenization. The errors are not idiosyncratic. Across the full 18×18 AttnDiff matrix (153 model pairs) we group models by the tokenizer they use and average the CKA score within and across groups. Pairs drawn from the same tokenizer class score 0.955 on average ($n = 34$, minimum 0.669), whereas pairs from different classes score 0.346 ($n = 119$, maximum 0.934); Table 25 gives the breakdown. The gap is not a proxy for lineage, because it survives when lineage is held fixed. LLaMA2-7B and Amber are two independently trained models that happen to tokenize the same way and share no weights: they score 0.9687. LLaMA2-7B and OpenLLaMA-7B are equally unrelated, but OpenLLaMA-7B uses the LLaMA-1 tokenizer: they score 0.2453, and Amber against OpenLLaMA scores 0.2156. Two pairs with the same architectural relationship and the same (absent) provenance differ by 0.72 purely as a function of tokenization. The mechanism is straightforward: the statistic is computed on a fixed prompt set, and two models that segment a prompt into the same tokens inherit correlated per-head structure from that shared segmentation; differencing a minimally corrupted prompt removes prompt-specific content but not this shared segmentation. Consequently the 18×18 matrix forms blocks by tokenizer, not by lineage.

Consequence for weight repackaging. On the documented repackaging pairs AttnDiff is high—Mistral-7B/Mistral-11B 0.9989, Solar-10.7B/Mistral-11B 0.9868, Mistral-7B/Solar-10.7B

Table 23: Labeled pair set for the calibrated evaluation, grouped by provenance. Both scores are reported for all 40 pairs.

Model pair	Label	FNF	REEF
LLaMA2-7B / LLaMA2-7B-chat	Deriv.	0.9351	0.9877
Qwen2.5-7B / Qwen2.5-7B-Instruct	Deriv.	0.9196	0.9225
LLaMA2-chat / Vicuna-v1.5	Deriv.	0.6463	0.7396
LLaMA2-chat / Chinesellama-2-7B	Deriv.	0.9454	0.9983
LLaMA2-chat / Codellama-7B	Deriv.	0.8667	0.8833
LLaMA2-chat / Llemma-7B	Deriv.	0.8593	0.9701
Mistral-7B / Solar-10.7B	Deriv.	0.5743	0.3226
Mistral-7B / Mistral-11B	Deriv.	0.5820	0.9923
Qwen2.5-7B-IT / Qwen1.5-7B-IT	Fam.	0.4043	0.7958
Qwen2.5-7B-IT / Qwen2.5-3B-IT	Fam.	0.6017	0.7988
Qwen2.5-7B-IT / Qwen2.5-1.5B-IT	Fam.	0.4705	0.6832
ChatGLM2-6B / ChatGLM3-6B	Fam.	0.5882	0.7475
Solar-10.7B / Mistral-11B	Fam.	0.7958	0.3320
Vicuna-7B / CodeLlama-7B	Fam.	0.6177	0.4700
Vicuna-7B / Llemma-7B	Fam.	0.6116	0.5989
Vicuna-7B / Chinese-LLaMA-2-7B	Fam.	0.6348	0.7458
CodeLlama-7B / Llemma-7B	Fam.	0.9081	0.9529
CodeLlama-7B / Chinese-LLaMA-2-7B	Fam.	0.8617	0.8934
Llemma-7B / Chinese-LLaMA-2-7B	Fam.	0.8516	0.9691
LLaMA2-7B / Amber	Indep.	0.3142	0.3673
LLaMA2-7B / OpenLLaMA-7B	Indep.	0.3757	0.4159
Qwen2.5-7B-IT / LLaMA3-8B	Indep.	0.2586	0.7944
Qwen2.5-7B-IT / ChatGLM3-6B	Indep.	0.1507	0.6842
Qwen2.5-7B-IT / GLM4-9B	Indep.	0.2737	0.8567
ChatGLM2-6B / LLaMA3-8B	Indep.	0.2612	0.9071
ChatGLM2-6B / GLM4-9B	Indep.	0.3626	0.8845
Qwen2.5-7B-IT / Vicuna-7B	Indep.	0.2311	0.3007
Qwen2.5-7B-IT / CodeLlama-7B	Indep.	0.1778	0.2679
Qwen2.5-7B-IT / Llemma-7B	Indep.	0.1800	0.2793
Qwen2.5-7B-IT / Chinese-LLaMA-2-7B	Indep.	0.1435	0.2779
Qwen2.5-3B-IT / Vicuna-7B	Indep.	0.2049	0.4284
Qwen2.5-3B-IT / CodeLlama-7B	Indep.	0.1797	0.3348
Qwen2.5-3B-IT / Llemma-7B	Indep.	0.1375	0.3357
Qwen2.5-3B-IT / Chinese-LLaMA-2-7B	Indep.	0.1257	0.3666
Qwen2.5-7B-IT / Mistral-7B	Indep.	0.1965	0.4544
Qwen2.5-3B-IT / Mistral-7B	Indep.	0.1743	0.4665
Vicuna-7B / Mistral-7B	Indep.	0.2046	0.0875
CodeLlama-7B / Mistral-7B	Indep.	0.1973	0.0194
Llemma-7B / Mistral-7B	Indep.	0.2108	0.0290
Chinese-LLaMA-2-7B / Mistral-7B	Indep.	0.1689	0.0304

Table 24: Per-pair comparison on the 36 pairs that all three methods support. The four ChatGLM pairs are excluded because AttnDiff cannot produce a fingerprint for ChatGLM-2 or ChatGLM-3. FNF entries above the largest independent score (0.3757) are in bold. FNF and REEF values are the same measurements as those behind Table 4, recomputed over this subset.

Pair	Provenance	FNF	REEF	AttnDiff
LLaMA2-chat / Vicuna-7B	Deriv.	**0.6463**	0.7396	0.9989
Mistral-7B / Mistral-11B	Deriv.	**0.5820**	0.9923	0.9989
LLaMA2-7B / LLaMA2-chat	Deriv.	**0.9351**	0.9877	0.9988
LLaMA2-chat / Chinese-LLaMA-2	Deriv.	**0.9454**	0.9983	0.9986
LLaMA2-chat / CodeLlama-7B	Deriv.	**0.8667**	0.8833	0.9915
Qwen2.5-7B / Qwen2.5-7B-IT	Deriv.	**0.9196**	0.9225	0.9913
LLaMA2-chat / Llemma-7B	Deriv.	**0.8593**	0.9701	0.9854
Mistral-7B / Solar-10.7B	Deriv.	**0.5743**	0.3226	0.9850
Vicuna-7B / Chinese-LLaMA-2	Fam.	**0.6348**	0.7458	0.9986
CodeLlama-7B / Chinese-LLaMA-2	Fam.	**0.8617**	0.8934	0.9931
Vicuna-7B / CodeLlama-7B	Fam.	**0.6177**	0.4700	0.9897
Solar-10.7B / Mistral-11B	Fam.	**0.7958**	0.3320	0.9868
Vicuna-7B / Llemma-7B	Fam.	**0.6116**	0.5989	0.9850
Llemma-7B / Chinese-LLaMA-2	Fam.	**0.8516**	0.9691	0.9816
CodeLlama-7B / Llemma-7B	Fam.	**0.9081**	0.9529	0.9797
Qwen2.5-7B-IT / Qwen2.5-3B-IT	Fam.	**0.6017**	0.7988	0.9670
Qwen2.5-7B-IT / Qwen1.5-7B-IT	Fam.	**0.4043**	0.7958	0.9061
Qwen2.5-7B-IT / Qwen2.5-1.5B-IT	Fam.	**0.4705**	0.6832	0.8427
LLaMA2-7B / Amber	Indep.	0.3142	0.3673	0.9687
Qwen2.5-7B-IT / GLM4-9B	Indep.	0.2737	0.8567	0.9340
Qwen2.5-7B-IT / LLaMA3-8B	Indep.	0.2586	0.7944	0.8739
CodeLlama-7B / Mistral-7B	Indep.	0.1973	0.0194	0.5536
Llemma-7B / Mistral-7B	Indep.	0.2108	0.0290	0.5482
Vicuna-7B / Mistral-7B	Indep.	0.2046	0.0875	0.5421
Chinese-LLaMA-2 / Mistral-7B	Indep.	0.1689	0.0304	0.5388
Qwen2.5-7B-IT / Mistral-7B	Indep.	0.1965	0.4544	0.3809
Qwen2.5-3B-IT / Mistral-7B	Indep.	0.1743	0.4665	0.3437
Qwen2.5-7B-IT / Llemma-7B	Indep.	0.1800	0.2793	0.2495
LLaMA2-7B / OpenLLaMA-7B	Indep.	0.3757	0.4159	0.2453
Qwen2.5-7B-IT / Vicuna-7B	Indep.	0.2311	0.3007	0.2213
Qwen2.5-7B-IT / Chinese-LLaMA-2	Indep.	0.1435	0.2779	0.2096
Qwen2.5-7B-IT / CodeLlama-7B	Indep.	0.1778	0.2679	0.2084
Qwen2.5-3B-IT / Llemma-7B	Indep.	0.1375	0.3357	0.1921
Qwen2.5-3B-IT / Vicuna-7B	Indep.	0.2049	0.4284	0.1740
Qwen2.5-3B-IT / CodeLlama-7B	Indep.	0.1797	0.3348	0.1679
Qwen2.5-3B-IT / Chinese-LLaMA-2	Indep.	0.1257	0.3666	0.1652

Table 25: AttnDiff score averaged within and across tokenizer classes over all 153 unordered pairs of the 18 models for which it can produce a fingerprint. "LLaMA-2" includes Vicuna-7B, Chinese-LLaMA-2-7B, CodeLlama-7B, Llemma-7B and Amber, which share the LLaMA-2 tokenizer.

Grouping	Pairs	Mean AttnDiff CKA
Within LLaMA-2 tokenizer class	21	0.984
Within Qwen tokenizer class	10	0.883
Within Mistral tokenizer class	3	0.990
Across tokenizer classes	119	0.346

0.9850—and all three are above every independent pair in our set, so on this benchmark it does flag the repackaged models. The margin is thin, however: the highest-scoring independent pair, LLaMA2-7B/Amber, reaches 0.9687, within 0.016 of the weakest repackaged pair, and it does so because Amber shares the LLaMA-2 tokenizer, not because any weights are shared. Every pair that shares the Mistral tokenizer is scored highly as well, including the same-family Solar-10.7B/Mistral-11B pair, so a high score is only weak evidence that what has been detected is repackaging rather than tokenization. We emphasise that this is a statement about transfer to *lineage verification*, not about AttnDiff's quality on the tasks for which it was designed: on answer-level model comparison with a tokenizer held fixed, and on its own benchmarks, it performs well. Our negative class, however, contains pairs that share a tokenizer while sharing no weights, and that is exactly the case its statistic does not separate.