RELIABLE LLM PROVENANCE VIA PANIC-PATTERN FINGERPRINTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training competitive large language models (LLMs) is expensive, making LLMs high-value intellectual property and motivating reliable provenance tests for both open-weight and API-served models. In this paper, we propose a unified provenance framework based on *panic patterns*—model-specific fingerprints elicited by stressful inputs. In the white-box setting, we introduce **P**anic **A**lignment **D**ifference (PAD), which fingerprints models by measuring stress-induced shifts in layerwise representation alignment via a clean-versus-adversarial differential. In the black-box setting, we introduce **O**pen-ended **Q**uestion **T**est (OQT), which uses open-ended questions as stressors and compares residual response embeddings after removing the consensus answer estimated from a reference pool. Across a range of open and proprietary LLMs, our stress-response fingerprints reduce spurious similarity, remain informative under the evaluated model variations, and provide complementary provenance evidence across access levels with bounded verification cost.

1 INTRODUCTION

Training competitive Large Language Models (LLMs) from scratch remains extremely costly in compute, data, and engineering (Maslej et al., 2025; Scao et al., 2022). Consequently, LLMs constitute high-value intellectual property (IP), and model owners increasingly rely on licensing and controlled distribution to regulate downstream use and protect ownership (Touvron et al., 2023). Yet models may be repackaged or fine-tuned while being claimed as independently trained, and proprietary endpoints may be imitated under black-box access (Carlini et al., 2024; Feng & Tramèr, 2024; Orekondy et al., 2019; Tramèr et al., 2016; Yao et al., 2024). These risks motivate reliable provenance methods across access levels.

Feature-based methods extract post-training signals such as weights or hidden representations and compare a suspect model with a reference (Wang et al., 2025; Zeng et al., 2024; Zhang et al., 2025a). These methods are non-intrusive, but weight fingerprints can be fragile under fine-tuning and pruning, while representation similarity on standard inputs can remain high for unrelated model families (Fernandez et al., 2024; Nikolic et al., 2025a). Watermarking instead embeds ownership signals through query–key triggers, backdoors, or controlled decoding (Gloaguen et al., 2025; Russinovich & Salem, 2024; Xu et al., 2024a). Such methods require intervention before release and may be removed or evaded once an adversary adapts the model or filters the trigger (Wang & Kerschbaum, 2019; Zhang et al., 2025b).

These limitations create a tension among *detectability*, *robustness*, and *practicality*. A useful provenance framework should separate related and unrelated models with few false positives, remain informative after common post-training modifications, have a bounded verification cost, and operate under both white-box and black-box access.

We address this problem by extracting fingerprints from models' *stress reactions* to unknown, underspecified, or misleading inputs. Such stressors elicit model-specific *panic patterns* that are relatively stable within a model family but differ across unrelated families. In the white-box setting, **Panic Alignment Difference (PAD)** measures how adversarial stress changes internal alignment geometry. In the black-box setting, **Open-ended Question Test (OQT)** embeds responses to underdetermined questions and removes the consensus answer of a verifier-held reference pool before comparing two models.

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.

The two methods provide evidence at different access levels. PAD supports family-level comparison under fine-tuning, pruning, and merging, while OQT provides a screening and ranking signal from API responses alone. We evaluate PAD on 34 checkpoints from nine model families and OQT on 35 models from 12 vendor lines. We additionally test detector-aware adaptation for PAD and reference-pool size, response post-processing, and in-context imitation for OQT.

2 Related Work

White-box provenance. Weight-based methods compare layer statistics, direct weight features, or matrix decompositions (Refael et al., 2024; Jia et al., 2022). They are data-free, but common post-training transformations such as extensive fine-tuning and pruning can weaken the fingerprint (Fernandez et al., 2024). Representation-based methods instead compare input-induced hidden features using CKA or gradient statistics (Wu et al., 2025; Zhang et al., 2025a). These signals are more dynamic than direct weight matching, but require internal access, add inference cost, and can retain spurious similarity when evaluated only on standard in-distribution inputs (Nikolic et al., 2025a).

Black-box provenance. Behavioral methods infer model identity from crafted prompts or from semantic and stylistic output statistics (Sam et al., 2025; Iourovitski et al., 2024; Pasquini et al., 2025). They suit API-only settings, but their signals depend on prompts and decoding and may be diluted by rewriting, paraphrasing, or response filtering (Xu et al., 2024a). Hand-crafted identification prompts can also make the attribution pipeline harder to justify or generalize (Pasquini et al., 2025). A related line uses hypothesis testing to ask whether two models' outputs are statistically distinguishable and treats indistinguishability as evidence of shared origin (Nikolic et al., 2025b). We compare OQT with both kinds of black-box evidence.

Watermarking. Query–key, backdoor, and controlled-decoding schemes embed a detectable signal in model behavior (Gloaguen et al., 2025; Russinovich & Salem, 2024; Xu et al., 2024a). They provide strong evidence when the signal survives, but require proactive changes to training or generation, are difficult to apply retroactively to released models, and can be weakened by adaptation, response filtering, or even character-level perturbations (Wang & Kerschbaum, 2019; Zhang et al., 2025b).

3 Preliminary

3.1 LLM Fingerprinting and Provenance

We view *LLM provenance* as identifying whether a suspect model is derived from a victim model or family through common development processes such as fine-tuning, pruning, and merging, rather than being independently trained. A *fingerprint* is a compact and reproducible signature that is stable within a model family but discriminative across unrelated families. Depending on the access level, it can be constructed from weights or activations in a white-box setting, or from model I/O alone in a black-box setting. Given models f and g, we compare their fingerprints to obtain evidence of a shared weight origin under a stated evaluation protocol.

For the black-box evaluation, a *lineage* pair shares a documented checkpoint, a *co-release* pair belongs to one vendor line without a documented shared checkpoint, and all remaining pairs are treated as *unrelated*. We report lineage and co-release detection separately.

3.2 Stressors

REEF compares representations induced by standard in-distribution inputs (Zhang et al., 2025a). Although effective in many cases, clean-input similarity can yield false positives because unrelated families still share broad representational structure (Nikolic et al., 2025a). Adversarial inputs instead probe decision boundaries, where model-specific differences can be more visible (Xu et al., 2024b). Replacing clean inputs with adversarial inputs preserves REEF's successful within-family comparison in Table 21 and reduces several cross-family similarities in Table 22, but adversarial inputs alone do not provide sufficient separation. PAD therefore compares the *change* from clean to adversarial conditions rather than either condition in isolation.

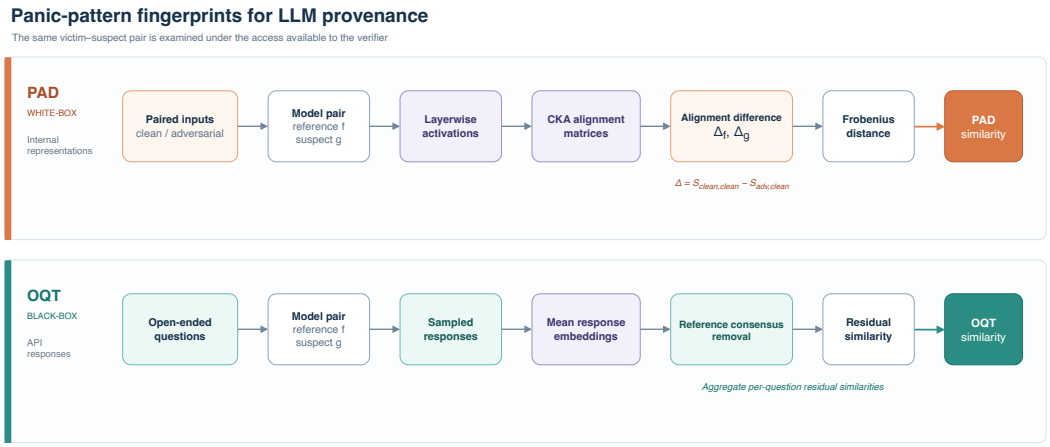

Figure 1: Overview of the panic-pattern framework. PAD uses paired clean and adversarial inputs to form a difference of layerwise alignment matrices and compares fingerprints with Frobenius distance. OQT uses open-ended questions, removes the reference-pool consensus from mean response embeddings, and aggregates residual similarities.

In the black-box setting, open-ended questions provide a different stressor. Prior work observes that models give highly homogeneous answers despite the task's intrinsic variability, both across models and across samples from one model (Anderson et al., 2024; Jiang et al., 2025; Wenger & Kenett, 2025; Yang & Zhao, 2024). This homogeneity makes absolute response similarity a weak provenance signal, because much of the answer is determined by content shared across well-trained models. OQT therefore estimates this consensus answer and compares the residual response embedding left after removing it.

4 Methodology

LLM provenance asks whether a suspect model descends from a reference lineage, but standard inputs are often dominated by shared pretraining priors, which can make unrelated models appear similar. The difficulty differs by access level: a white-box verifier can inspect representations but must tolerate post-training modifications, while a black-box verifier observes only stochastic outputs that are already homogeneous across many models. We use stressors to expose model-dependent behavior that standard inputs conceal.

As illustrated in Figure 1, PAD and OQT instantiate this idea with the information available to the verifier. PAD measures a clean–adversarial change in internal representation geometry. OQT uses underdetermined questions and removes a reference-pool consensus before comparing response embeddings. Both return continuous similarity evidence; the operating interpretation is defined by each evaluation protocol.

4.1 Panic-induced Differential Alignment

Our preliminary comparison shows that adversarial inputs reduce some spurious REEF similarities, but the remaining separation is insufficient. Clean and adversarial evaluations can both contain a lineage-irrelevant background component. PAD is designed to suppress this component through a clean–adversarial differential.

For a model, we define its panic pattern as the change in internal representation geometry from clean inputs to paired adversarial stressors. PAD extracts layerwise activations under both conditions, summarizes each condition with a layerwise alignment matrix, and subtracts the clean baseline from the stress condition. The resulting difference-of-alignments fingerprint records where and how strongly the geometry changes, rather than the absolute geometry shared across models (Algorithm 1).

For aligned pairs $\{(x_i^{(c)}, x_i^{(a)})\}_{i=1}^m$ and layer l of model f, let

$$X_{f,l}^{(d)} \in \mathbb{R}^{m \times p_{f,l}}, \qquad \big(X_{f,l}^{(d)}\big)_{i,:} = \phi_{f,l}\big(x_i^{(d)}\big)^\top, \quad d \in \{c, a\}.$$

We instantiate the alignment operator A as linear CKA (Kornblith et al., 2019), which handles different feature dimensions and is invariant to orthogonal transformations and uniform scaling. The PAD fingerprint is

$$S_f^{(c,c)}[l, l'] = A\big(X_{f,l}^{(c)}, X_{f,l'}^{(c)}\big), \qquad S_f^{(a,c)}[l, l'] = A\big(X_{f,l}^{(a)}, X_{f,l'}^{(c)}\big), \tag{1}$$

$$\Delta_f = S_f^{(c,c)} - S_f^{(a,c)}. \tag{2}$$

We compare Δ_f and Δ_g with Frobenius distance. When the models have different numbers of layers, we first align the two matrices to a common grid with bilinear interpolation. A smaller distance indicates more similar panic patterns; for reporting, we use the corresponding bounded PAD similarity score. Appendix B gives the CKA definition and kernel comparison, while Appendix A records the evaluated checkpoints and execution setup.

4.2 Open-ended Question Test

OQT uses open-ended questions as black-box stressors and fingerprints a model with the semantic embeddings of its responses. The questions are deliberately underdetermined: the prompt fixes the topic, while the model's training and post-training behavior influence which answer it chooses and how it expresses that answer. The main obstacle is the large amount of content shared across models. OQT estimates this consensus separately for each question and removes it before comparing a victim and suspect.

For model f, question q, and embedding model e, let $\mu_f(q) = \mathbb{E}[e_f(q)]$, estimated by the mean of repeated response embeddings. Given a verifier-held reference pool R, OQT computes

$$\bar{\mu}(q) = \frac{1}{|R|} \sum_{h \in R} \mu_h(q), \qquad \tilde{\mu}_f(q) = \mu_f(q) - \bar{\mu}(q), \qquad \mathrm{Sim}_q(f, g) = \cos\big(\tilde{\mu}_f(q), \tilde{\mu}_g(q)\big).$$

The vectors $\tilde{\mu}_f(q)$ and $\tilde{\mu}_g(q)$ represent how the two models depart from the reference consensus on the same question. The fixed-pool result averages their per-question cosine similarities and does not require the suspect's vendor line. We also report a line-held-out diagnostic that excludes the compared vendor lines when forming the consensus and applies a light lower-tail winsorization before the mean. This diagnostic avoids allowing a pair to contribute to its own reference consensus, but requires declared vendor-line metadata and is therefore not the default fixed-pool setting. Full estimator, threat model, elicitation conditions, and collection details are in Appendix D.

5 Experiment

We organize the evaluation around two questions: (1) does each method separate related from unrelated models, and (2) does its fingerprint remain informative under the evaluated model variations? PAD answers these questions with internal representations; OQT uses API responses only. We report the relationship definition, model cohort, and score protocol for each setting because their access assumptions differ.

5.1 Panic Alignment Difference

Setup. We form $m = 200$ clean–adversarial pairs from TruthfulQA (Lin et al., 2022) using the imperceptible-perturbation procedure (Boucher et al., 2022). The unified study contains 34 checkpoints from nine model families and all 561 unordered pairs: 79 within-family and 482 cross-family pairs. Logits (Yang & Wu, 2025), REEF (Zhang et al., 2025a), and Adv-REEF (REEF on adversarial inputs) are the baselines. Appendix A lists the checkpoints and executed setup.

Baselines. Logits fingerprints a model through its output space (Yang & Wu, 2025). REEF compares hidden representations induced by clean inputs (Zhang et al., 2025a); Adv-REEF uses the same comparison on the adversarial inputs. These baselines separate the effect of changing the input from PAD's clean–adversarial differential.

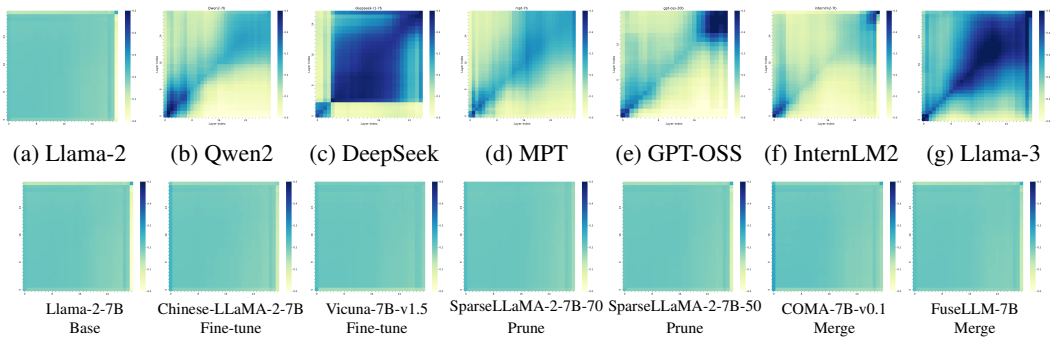

Figure 2: PAD fingerprints. Top: representative model families. Bottom: Llama-2 and variants obtained by fine-tuning, pruning, and merging.

Table 1: PAD similarity with MPT-7B as victim. The first model is an in-family derivative; the others are cross-family. Cell colors indicate similarity: > 0.8, 0.5–0.8, < 0.5.

Method	MPT-Inst.	Llama-2	Llama-3	Qwen2	DeepSeek	GPT-OSS	InternLM2
Logits	0.9999	0.5767	0.0000	0.0800	0.6467	0.0033	0.9999
REEF	0.9490	0.1709	0.4853	0.7436	0.3349	0.6909	0.8304
Adv-REEF	0.9320	0.1366	0.4662	0.6573	0.2122	0.6299	0.6687
PAD	0.9562	0.0205	0.0001	0.3006	0.0000	0.0455	0.0103

Effectiveness and robustness. Across all 561 pairs, PAD obtains AUROC 0.9404, AP 0.9240, family-macro AUROC 0.9256, and correct-family Top-1 retrieval for 23 of 25 non-reference checkpoints. These metrics use all 79 within-family and 482 cross-family pairs rather than a single chosen victim. Complete pairwise matrices and the supplementary SSIM matcher are in Appendix A.2.

Table 1 retains the MPT-7B comparison as a concrete example. PAD assigns MPT-7B-Instruct a similarity of 0.9562 and every cross-family candidate at most 0.3006. REEF, in contrast, assigns MPT-7B and InternLM2-7B a cross-family similarity of 0.8304, while PAD assigns 0.0103. This is a reference-specific observation rather than evidence that PAD universally outperforms REEF; the complete-cohort metrics characterize PAD across all nine families.

For robustness, we fix Llama-2-7B as the reference and compare fine-tuned, pruned, and merged variants. Their fingerprints retain the same visual structure in Figure 2; all evaluated similarities exceed 0.85, with mean 0.9121. The result supports a family-level signal under these common post-training modifications.

Adaptive attack. The preceding experiment considers ordinary post-training changes. A stronger attacker may know PAD and deliberately adapt a derived model to reduce its similarity to the reference while preserving utility. We therefore optimize rank-8 LoRA adapters on Llama-2-7B for 200 updates with a utility-regularized objective that reduces PAD similarity. A utility-only control uses the same initialization, data, and minibatch order. The attack uses 120 probe pairs for optimization, 40 development pairs for coefficient selection, and 40 question-disjoint test pairs for reporting.

Table 2: PAD-targeted adaptation on Llama-2-7B. The lower two rows report a separately recorded stress test; ARC-C and TQA-Gen denote ARC-Challenge and TruthfulQA-Gen.

Run	PAD sim.	PPL/base	MCQ	MMLU	ARC-C	TQA-Gen
Utility-only control	0.9840	0.8657	28/64	–	–	–
Controlled PAD-aware	0.9432	0.8924	27/64	–	–	–
Base (separate test)	–	–	–	0.41	0.43	0.33
PAD-targeted variant	0.79	–	–	0.23	0.21	0.01

After 200 updates, the PAD-aware run has a similarity score of 0.9432, compared with 0.9840 for the matched utility-only control. The utility-only and PAD-aware models answer $28/64$ and $27/64$

questions correctly and obtain PPL/base ratios of 0.8657 and 0.8924, respectively. Thus, targeting PAD produces a measurable score difference of 0.0408, while the two recorded utility diagnostics remain comparable.

The separately recorded stress test reaches a lower PAD score of 0.79, but its MMLU, ARC-Challenge, and TruthfulQA-Generation scores fall from 0.41, 0.43, and 0.33 to 0.23, 0.21, and 0.01, respectively (Table 2). Together, the runs show a limited score change under the controlled budget and a substantial utility cost in the run reaching a lower score. Both results cover one base model and do not establish universal attack resistance; Appendix A.3 gives the full trajectory and configuration.

5.2 Open-ended Question Test

Setup. **Models and ground truth.** We evaluate 35 models from 12 vendor lines, chosen to cover the three relations defined in subsection 3.1: 12 checkpoint-sharing lineage pairs, 66 co-release pairs from one vendor line without a documented shared checkpoint, and 517 unrelated pairs. The cohort spans dense and Mixture-of-Experts architectures, 1B–120B parameters, open-weight and proprietary endpoints, and models with and without visible chain of thought. The complete registry is provided in Table 17.

Questions and collection. We sample 75 open-ended questions (Cao & Wang, 2021) and present them in the same order to every model. We target ten responses per question and model, remove displayed deliberation traces from reasoning-model responses, and use Qwen-Embedding-8B for the main results. The verifier holds reference fingerprints and has black-box access to a suspect endpoint; the endpoint does not see the question set in advance. Appendix D gives the full threat model, response accounting, and estimator settings.

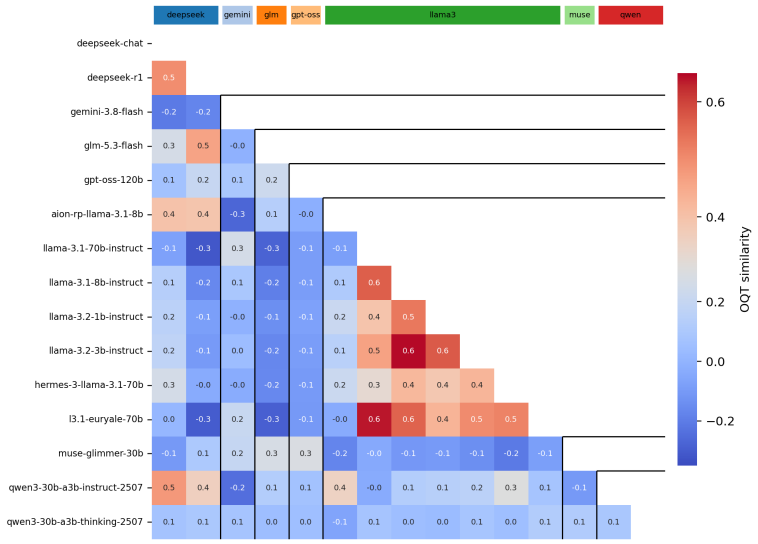

Figure 3: Representative OQT score matrix for 15 of the 35 models under the line-held-out diagnostic. Black boundaries separate vendor lines; one triangle is shown because the score is symmetric. The complete matrix is shown in Figure 9.

Table 3: Core OQT results on the 595 evaluated pairs. The line-held-out row is a metadata-assisted diagnostic; the fixed-reference row uses one common reference consensus.

Score	Lineage AUC	Top-1	Mean rank	Co-release AUC
Absolute response similarity	0.644	3/12	10.0	0.549
Residual, fixed reference (mean)	0.840	3/12	5.7	0.689
Residual, line held out (diagnostic)	0.850	4/12	5.8	0.697

5.2.1 RQ1: EFFECTIVENESS

Absolute and residual scores. We first test whether absolute response similarity already measures lineage. It does not do so reliably in this cohort: checkpoint-sharing and unrelated pairs have median absolute similarities of 0.878 and 0.848, and the highest-scoring pair in the matrix is unrelated (0.947). As shown in Table 3, absolute similarity yields lineage ROC-AUC 0.644. Removing the fixed reference consensus raises the ROC-AUC to 0.840 and improves mean retrieval rank from 10.0 to 5.7 among 34 candidates. This comparison uses the same responses and changes only the score construction.

Lineage retrieval. The line-held-out diagnostic reaches ROC-AUC 0.850 and ranks a checkpoint-sharing partner first for 4 of 12 victims, compared with 3 of 12 for absolute similarity. Its mean partner rank is 5.8 of 34, and no lineage partner falls below rank 18. At the threshold that maximizes F_1 on these scores, it identifies 6 of the 12 lineage pairs and includes 14 of the 517 unrelated pairs ($F_1 = 0.375$); the absolute score identifies 5 lineage pairs and includes 37 unrelated pairs. Figure 3 shows the representative matrix, and Figure 9 provides the full 35-model result.

The remaining errors are concentrated in behavior-retargeted derivatives: roleplay fine-tunes and an instruction/reasoning pair rank lower than derivatives that retain the base model's disposition. This behavior is consistent with the black-box score measuring response behavior rather than internal weight similarity, so we use it as continuous screening and ranking evidence.

Co-release pairs and aggregation. Co-release detection is weaker: the fixed-pool score obtains ROC-AUC 0.689, compared with 0.549 for absolute similarity, while the line-held-out diagnostic obtains 0.697. These pairs share a vendor line but not a documented checkpoint, and their scores overlap substantially with unrelated pairs. We therefore report co-release separately rather than combining it with checkpoint lineage. Table 16 compares aggregation choices; mean, trimmed mean, winsorized mean, and median are close, while rank and mixture aggregation are weaker on the lineage task.

5.2.2 BLACK-BOX BASELINES

We compare OQT against the direct comparison of its own responses, a kernel two-sample MMD test (Gretton et al., 2012), the hypothesis-testing provenance method of Nikolic et al. (2025b), LLMmap (Pasquini et al., 2025), and a supervised pair classifier. Table 4 uses the relaxed same-vendor-line task, which includes both lineage and co-release pairs and is therefore distinct from the checkpoint-lineage task in Table 3. All rows except LLMmap reuse OQT's responses; LLMmap is run from the authors' released code with its own identification queries. Appendix N records the interface substitutions and query budgets.

Table 4: Black-box comparison on the relaxed same-vendor-line task. Best F_1 is the oracle operating point for each score.

Method	**ROC-AUC**	**Best F_1**
OQT (ours)	0.732	0.440
Direct response similarity	0.567	0.251
MMD two-sample test (Gretton et al., 2012)	0.556	0.247
Nikolic et al. (Nikolic et al., 2025b)	0.709	0.369
LLMmap (Pasquini et al., 2025)	**0.855**	**0.586**
Supervised pair classifier	0.657	0.313

On the same pair population and responses, OQT reaches ROC-AUC 0.732, exceeding direct similarity (0.567), MMD (0.556), and hypothesis testing (0.709). LLMmap is strongest at 0.855, but uses identification queries for a closed-set classifier. The supervised pair classifier reaches 0.657, indicating that OQT's gain over direct comparison is not explained by adding a learned relation classifier. OQT is therefore the strongest evaluated label-free method using the shared open-ended response collection, while LLMmap remains stronger overall.

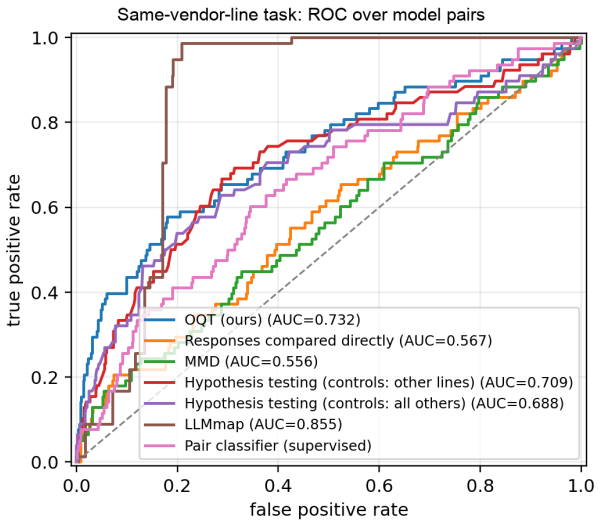

Figure 4: ROC curves for black-box methods on the relaxed same-vendor-line task. This task and pair population match Table 4; it differs from the stricter checkpoint-lineage task in Table 3.

5.2.3 RQ2: Robustness and efficiency

We test reference-pool size, response post-processing, and in-context imitation in Appendix J. Lineage AUROC rises from 0.688 with two references to 0.815 with 16 and reaches 0.812 with the full eligible pool, showing that performance is poor with a very small pool but stabilizes before every reference is used. Truncating answers to two sentences and prepending a politeness prefix yield AUROC 0.886 and 0.845, compared with 0.838 in the matched control; rewriting responses with an unrelated model yields 0.777.

We also recompute the relaxed same-vendor-line task with five alternative embedding models. Across all six embedders, ROC-AUC ranges from 0.716 to 0.791, and the alternative rankings correlate with the main result at 0.908–0.988. Their optimal thresholds vary from 0.237 to 0.368, which supports fixing the embedder before calibration. Full results are in Table 10.

For a detector-aware black-box attack, we place examples of a victim's answers in the suspect model's context. At 30 imitation examples, the victim ranks first in two of six evaluated pairs and four pairs exceed the relaxed co-release threshold; two of those four already exceeded it without imitation. The result shows that imitation can move a score and also illustrates why OQT assumes that the verifier's questions are not revealed in advance.

Efficiency. The main collection targets 750 responses per model (75 questions times 10 samples). In the recorded expensive cases, collection costs range from \$0.20 to \$1.30 per model, with 76K–873K output tokens; reasoning and long-response models dominate this cost. Once fingerprints are collected, pairwise comparison and aggregation are negligible matrix operations. Complete request, token, failure, and cost accounting is in Appendix K.

5.2.4 Decision boundary

Similarity is continuous: a lightly fine-tuned derivative can remain close to its base, while behavior-retargeting post-training can move it farther away. A decision threshold also depends on the embedding model, question set, reference pool, and the different costs of a false accusation and a missed violation. We therefore calibrate on reference pairs excluded from evaluation rather than present one universal constant.

In the reported split, a threshold calibrated on three vendor lines obtains $F_1 = 0.586$ on calibration pairs but transfers poorly to held-out lines: precision is 0.057 at recall 0.667 on the co-release task. An operating point with true-positive rate 0.95 has false-positive rate 0.923. Appendix I reports the

full calibration and trade-off. We consequently use ranking metrics as the primary OQT evidence and report thresholded results only together with their calibration protocol.

5.3 Discussion and Limitations

PAD isolates stress-induced representation changes through a clean–adversarial differential. Across the evaluated cohort, this suppresses cross-family similarities that remain high for clean-input representation matching and preserves a recognizable signal after fine-tuning, pruning, and merging. Its current Frobenius comparison nevertheless remains sensitive to entry-wise magnitude: two fingerprints can share structure while differing in scale. The supplementary SSIM analysis evaluates a structure-aware matcher without changing the PAD fingerprint or the primary method.

OQT addresses a different obstacle. Open-ended answers contain a large consensus component, so absolute response similarity can be high for unrelated models. Removing a reference-pool consensus improves checkpoint-lineage ranking on the evaluated cohort, but the remaining signal is behavioral: independently trained co-releases and derivatives whose post-training retargets behavior remain difficult. The fixed pool avoids needing the suspect's vendor metadata, while the stronger line-held-out result is only a diagnostic for cases where that metadata is available. Accordingly, PAD and OQT provide different forms of evidence, and their numerical scores are not interchangeable.

Both evaluations use documented relationships as ground truth rather than weight-level verification, so an incorrect registry entry becomes a labeling error. The PAD adaptive study covers one base model, one optimization seed, and one LoRA parameterization. OQT scores and thresholds require recalibration for a new embedding model, question set, reference pool, or model generation. OQT also assumes that the verifier's questions are not exposed before collection. We test fine-tuning, response post-processing, and in-context imitation, but not full retraining or distillation by an adversary who observes the verifier. These constraints limit the conclusions to the reported protocols.

6 Conclusion

This paper proposes *panic-pattern fingerprinting* as a unified provenance signal for LLM lineage verification. In the white-box setting, **PAD** fingerprints models via stress-induced shifts in layerwise representation alignment through a clean–adversarial differential. In the black-box setting, **OQT** compares residual response embeddings after removing the consensus answer of a reference pool. Across 34 checkpoints from nine families, PAD remains informative under the evaluated fine-tuning, pruning, merging, and detector-aware adaptation settings. Across 35 models from 12 vendor lines, OQT improves checkpoint-lineage ranking over absolute response similarity, while its calibration analysis highlights the need for protocol-specific thresholds. Together, the two methods provide complementary provenance evidence across access levels and remain informative under the evaluated model variations.

ETHICS STATEMENT

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

AI USE STATEMENT

In preparing this work, we used OpenAI Codex to assist with writing and language polishing, retrieval and discovery of relevant literature, and drafting sections of the manuscript. Codex also supported discussions of research plans and methodological options, code development, and experimental execution. The authors have reviewed and verified the AI-assisted content. The authors take full responsibility for the final manuscript, including its text, references, methods, experimental results, and conclusions.

REPRODUCIBILITY STATEMENT

The PAD and OQT procedures are specified in Section 4, with pseudocode in Appendix S. Section 5 describes the evaluated models, datasets, and experimental settings. Appendix A provides the PAD checkpoint registry and executed setup, while Appendix A.3 records the controlled adaptive-attack configuration and trajectory. Appendix B details the alignment operator. Appendices D, E, F, G, I, J, and N describe the revised OQT estimator, implementation, pruning, embedding sensitivity, threshold analysis, robustness studies, and baselines.

REFERENCES

Barrett R. Anderson, Jash Hemant Shah, and Max Kreminski. Homogenization effects of large language models on human creative ideation. In *Creativity & Cognition*, pp. 413–425. ACM, 2024.

Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. Bad characters: Imperceptible NLP attacks. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pp. 1987–2004. IEEE, 2022. doi: 10.1109/SP46214.2022.9833641. URL https://doi.org/10.1109/SP46214.2022.9833641.

Shuyang Cao and Lu Wang. Controllable open-ended question generation with a new question type ontology. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6424–6439, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.502. URL https://aclanthology.org/2021.acl-long.502.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Eric Wallace, David Rolnick, and Florian Tramèr. Stealing part of a production language model. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=VE3yWXt3KB.

Shanglun Feng and Florian Tramèr. Privacy backdoors: Stealing data with corrupted pretrained models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=7yixJXmzb8.

Pierre Fernandez, Guillaume Couairon, Teddy Furon, and Matthijs Douze. Functional invariants to watermark large transformers. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*, pp. 4815–4819. IEEE, 2024. doi: 10.1109/ICASSP48485.2024.10447264. URL https://doi.org/10.1109/ICASSP48485.2024.10447264.

Thibaud Gloaguen, Robin Staab, Nikola Jovanovic, and Martin T. Vechev. Robust LLM fingerprinting via domain-specific watermarks. *CoRR*, abs/2505.16723, 2025. doi: 10.48550/ARXIV.2505.16723. URL https://doi.org/10.48550/arXiv.2505.16723.

Arthur Gretton, Olivier Bousquet, Alexander J. Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In Sanjay Jain, Hans Ulrich Simon, and Etsuji Tomita (eds.), *Algorithmic Learning Theory, 16th International Conference, ALT 2005, Singapore, October 8-11, 2005, Proceedings*, volume 3734 of *Lecture Notes in Computer Science*, pp. 63–77. Springer, 2005. doi: 10.1007/11564089_7. URL https://doi.org/10.1007/11564089_7.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012. URL https://jmlr.org/papers/v13/gretton12a.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

Dmitri Iourovitski, Sanat Sharma, and Rakshak Talwar. Hide and seek: Fingerprinting large language models with evolutionary learning. *CoRR*, abs/2408.02871, 2024. doi: 10.48550/ARXIV.2408.02871. URL https://doi.org/10.48550/arXiv.2408.02871.

Hengrui Jia, Hongyu Chen, Jonas Guan, Ali Shahin Shamsabadi, and Nicolas Papernot. A zest of LIME: towards architecture-independent model distances. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=OUz_9TiTv9j.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, Alon Albalak, and Yejin Choi. Artificial hivemind: The open-ended homogeneity of language models (and beyond), 2025. URL https://arxiv.org/abs/2510.22954.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019. URL http://proceedings.mlr.press/v97/kornblith19a.html.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.229. URL https://doi.org/10.18653/v1/2022.acl-long.229.

Nestor Maslej, Loredana Fattorini, C. Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, Tobi Walsh, Armin Hamrah, Lapo Santarlasci, Julia Betts Lotufo, Alexandra Rome, Andrew Shi, and Sukrut Oak. Artificial intelligence index report 2025. *CoRR*, abs/2504.07139, 2025. doi: 10.48550/ARXIV.2504.07139. URL https://doi.org/10.48550/arXiv.2504.07139.

Ivica Nikolic, Teodora Baluta, and Prateek Saxena. Model provenance testing for large language models. *CoRR*, abs/2502.00706, 2025a. doi: 10.48550/ARXIV.2502.00706. URL https://doi.org/10.48550/arXiv.2502.00706.

Ivica Nikolic, Teodora Baluta, and Prateek Saxena. Model provenance testing for large language models. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025b. URL https://arxiv.org/abs/2502.00706.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 4954–4963. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00509. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Orekondy_Knockoff_Nets_Stealing_Functionality_of_Black-Box_Models_CVPR_2019_paper.html.

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.

Yehonathan Refael, Adam Hakim, Lev Greenberg, Tal Aviv, Satya Lokam, Ben Fishman, and Shachar Seidman. SLIP: securing llms IP using weights decomposition. *CoRR*, abs/2407.10886, 2024. doi: 10.48550/ARXIV.2407.10886. URL https://doi.org/10.48550/arXiv.2407.10886.

Mark Russinovich and Ahmed Salem. Hey, that's my model! introducing chain & hash, an LLM fingerprinting technique. *CoRR*, abs/2407.10887, 2024. doi: 10.48550/ARXIV.2407.10887. URL https://doi.org/10.48550/arXiv.2407.10887.

Dylan Sam, Marc Finzi, and J. Zico Kolter. Predicting the performance of black-box llms through self-queries. *CoRR*, abs/2501.01558, 2025. doi: 10.48550/ARXIV.2501.01558. URL https://doi.org/10.48550/arXiv.2501.01558.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022. doi: 10.48550/ARXIV.2211.05100. URL https://doi.org/10.48550/arXiv.2211.05100.

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, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In Thorsten Holz and Stefan Savage (eds.), *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*, pp. 601–618. USENIX Association, 2016. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer.

Suqing Wang, Ziyang Ma, Xinyi Li, and Zuchao Li. Ghost in the transformer: Tracing LLM lineage with svd-fingerprint. *CoRR*, abs/2511.06390, 2025. doi: 10.48550/ARXIV.2511.06390. URL https://doi.org/10.48550/arXiv.2511.06390.

Tianhao Wang and Florian Kerschbaum. Attacks on digital watermarks for deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pp. 2622–2626. IEEE, 2019. doi: 10.1109/ICASSP.2019.8682202. URL https://doi.org/10.1109/ICASSP.2019.8682202.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13 (4):600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://ece.uwaterloo.ca/~z70wang/publications/ssim.html.

Emily Wenger and Yoed Kenett. We're different, we're the same: Creative homogeneity across llms, 2025. URL https://arxiv.org/abs/2501.19361.

Zehao Wu, Yanjie Zhao, and Haoyu Wang. Gradient-based model fingerprinting for LLM similarity detection and family classification. *CoRR*, abs/2506.01631, 2025. doi: 10.48550/ARXIV.2506.01631. URL https://doi.org/10.48550/arXiv.2506.01631.

Jiashu Xu, Fei Wang, Mingyu Derek Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 3277–3306. Association for Computational Linguistics, 2024a. doi: 10.18653/V1/2024.NAACL-LONG.180. URL https://doi.org/10.18653/v1/2024.naacl-long.180.

Tianlong Xu, Chen Wang, Gaoyang Liu, Yang Yang, Kai Peng, and Wei Liu. United we stand, divided we fall: Fingerprinting deep neural networks via adversarial trajectories. *Advances in Neural Information Processing Systems*, 37:69299–69328, 2024b.

Dongjie Yang and Hai Zhao. Are llms aware that some questions are not open-ended? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 2142–2152. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.117. URL https://doi.org/10.18653/v1/2024.findings-emnlp.117.

Zhiguang Yang and Hanzhou Wu. A fingerprint for large language models, 2025. URL https://arxiv.org/abs/2407.01235.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm-v: A GPT-4V level MLLM on your phone. *CoRR*, abs/2408.01800, 2024. doi: 10.48550/ARXIV.2408.01800. URL https://doi.org/10.48550/arXiv.2408.01800.

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.

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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=SnDmPkOJ0T.

Zhaoxi Zhang, Xiaomei Zhang, Yanjun Zhang, He Zhang, Shirui Pan, Bo Liu, Asif Qumer Gill, and Leo Yu Zhang. Character-level perturbations disrupt LLM watermarks. *CoRR*, abs/2509.09112, 2025b. doi: 10.48550/ARXIV.2509.09112. URL https://doi.org/10.48550/arXiv.2509.09112.

A PAD EXPERIMENTAL DETAILS

A.1 UNIFIED COHORT AND EVALUATION SETUP

Cohort and relationship labels. The unified PAD study contains 34 checkpoints from nine model families with a documented common weight origin within each family. It includes instruction and domain fine-tuning, continued pretraining, model merging, pruning, and distillation. Positive pairs share a documented weight origin, including indirect descendants and sibling branches; all cross-family pairs are treated as negatives under this registry. These labels do not rule out weaker relationships through shared data or teacher models. The complete evaluation contains 79 positive and 482 negative unordered pairs. Table 5 lists every checkpoint used in this study.

Executed setup. Every checkpoint receives the same ordered 200 clean–adversarial rows derived from 179 source questions. Inference uses each checkpoint's own tokenizer, BF16, batch size 4, and one NVIDIA A100 40GB GPU. We record the last-valid-token output from every transformer block and apply the same preprocessing to every checkpoint. The executed environment uses PyTorch 2.10.0, Transformers 5.4.0, and NumPy 1.26.2.

Evaluation records. The main Frobenius matcher and the complementary SSIM matcher are evaluated on the same fingerprints and the same 561 model pairs. We report pooled AUROC and AP, family-macro AUROC, and reference-retrieval Top-1 accuracy.

A.2 COMPLETE COHORT RESULTS

Supplementary structural matching. Frobenius distance is the primary PAD matcher used in the main paper. For completeness, we additionally evaluate structural similarity (SSIM) (Wang et al., 2004) on the same fixed PAD fingerprints and the same 561 model pairs. This supplementary analysis changes only the matching function; the checkpoint cohort, fingerprints, and relationship labels remain unchanged.

Figures 5 and 6 visualize the complete pairwise score matrices underlying Table 6. Both matchers reveal family-level block structure, while SSIM produces sharper separation for many families. The raw-pruned SmolLM checkpoint (M22) remains a difficult case.

Table 5: Checkpoint registry for the unified PAD study. Daggers mark the nine reference checkpoints. Parameter counts refer to the loaded checkpoints.

ID	Checkpoint	Root	Params (M)	Layers	Transformation
M01	MPT-7B†	R1	6649.3	32	base
M02	Japanese-MPT-7B	R1	6649.3	32	CT
M03	MPT-7B-Chat	R1	6649.3	32	FT
M04	MPT-7B-Instruct	R1	6649.3	32	FT
M05	MPT-7B-Mercury	R1	6648.7	32	FT
M06	MPT-7B-Storywriter	R1	6649.3	32	CT
M07	Neural-Chat-7B-v1-1	R1	6648.7	32	FT
M08	Llama-2-7B†	R2	6738.4	32	base
M09	Chinese-LLaMA-2-7B	R2	6929.3	32	CT+vocab
M10	CodeLlama-7B DARE-TIES	R2	6738.4	32	merge
M11	CodeLlama-7B-Instruct	R2	6738.5	32	CT+FT
M12	COMA-7B-v0.1	R2	6738.4	32	merge
M13	Llemma-7B	R2	6738.5	32	CT
M14	Sheared-LLaMA-1.3B	R2	1345.4	24	prune+CT
M15	Sheared-LLaMA-2.7B	R2	2701.6	32	prune+CT
M16	Vicuna-7B-v1.5	R2	6738.4	32	FT
M17	WizardMath-7B-V1.0	R2	6738.4	32	FT+RL
M18	Pythia-70M†	R3	70.4	6	base
M19	Lamini-docs fine-tune	R3	70.4	6	FT
M20	SmolLM-135M†	R4	134.5	30	base
M21	SmolLM-135M-Instruct	R4	134.5	30	FT
M22	SmolLM-Instruct (raw-pruned)	R4	99.1	20	raw prune
M23	GPT-2†	R5	124.4	12	base
M24	DistilGPT2	R5	81.9	6	distill
M25	GPT-2-IMDb	R5	124.4	12	FT
M26	Qwen2.5-0.5B†	R6	494.0	24	base
M27	Qwen2.5-0.5B-Instruct	R6	494.0	24	post-train
M28	BLOOM-560M†	R7	559.2	24	base
M29	BLOOMZ-560M	R7	559.2	24	FT
M30	OPT-350M†	R8	331.2	24	base
M31	Pygmalion-350M	R8	331.2	24	FT
M32	Llama-3.1-8B-Instruct†	R9	8030.3	32	FT
M33	Llama-3.2-1B-Instruct	R9	1235.8	16	prune+KD
M34	Llama-3.2-3B-Instruct	R9	3212.7	28	prune+KD

R1: MPT-7B; R2: Llama-2-7B; R3: Pythia-70M; R4: SmolLM-135M; R5: GPT-2; R6: Qwen2.5-0.5B; R7: BLOOM-560M; R8: OPT-350M; R9: Llama-3.1-8B. CT: continued or domain pretraining; FT: fine-tuning; RL: reinforcement learning; KD: knowledge distillation.

Table 6: Complete-cohort PAD results. Frobenius is the primary matcher; SSIM is reported as a supplementary structural matcher.

Matcher	**AUROC**	**AP**	**Macro AUROC**	**Top-1**
Frobenius	0.9404	0.9240	0.9256	23/25
SSIM	0.9704	0.9505	0.9342	24/25

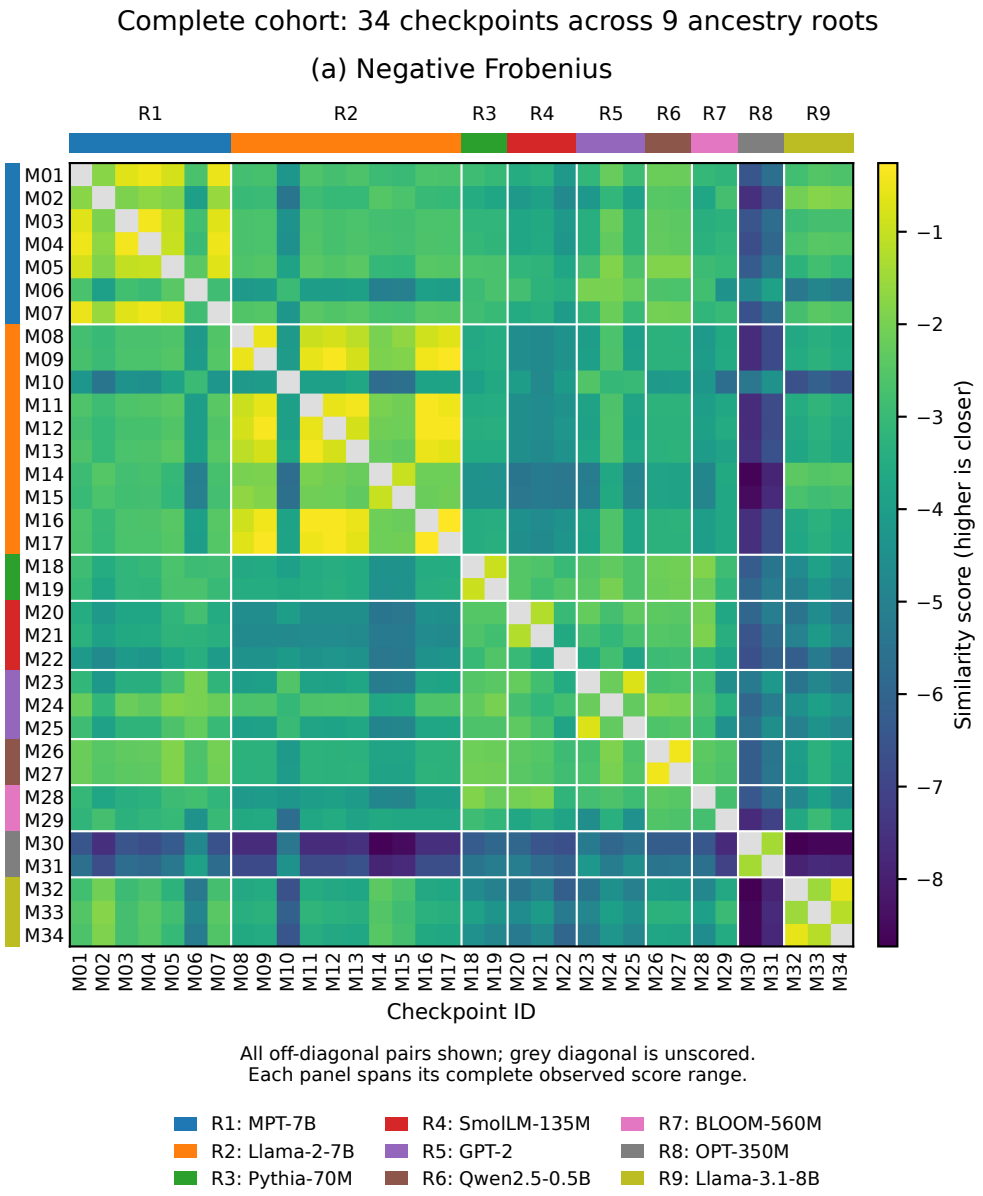

Figure 5: Complete-cohort Frobenius scores covering all 561 unordered pairs among the 34 checkpoints. We plot the negative Frobenius distance, so higher values indicate closer PAD fingerprints. Checkpoint IDs follow Table 5; white lines delimit the nine model families, and diagonal cells are unscored.

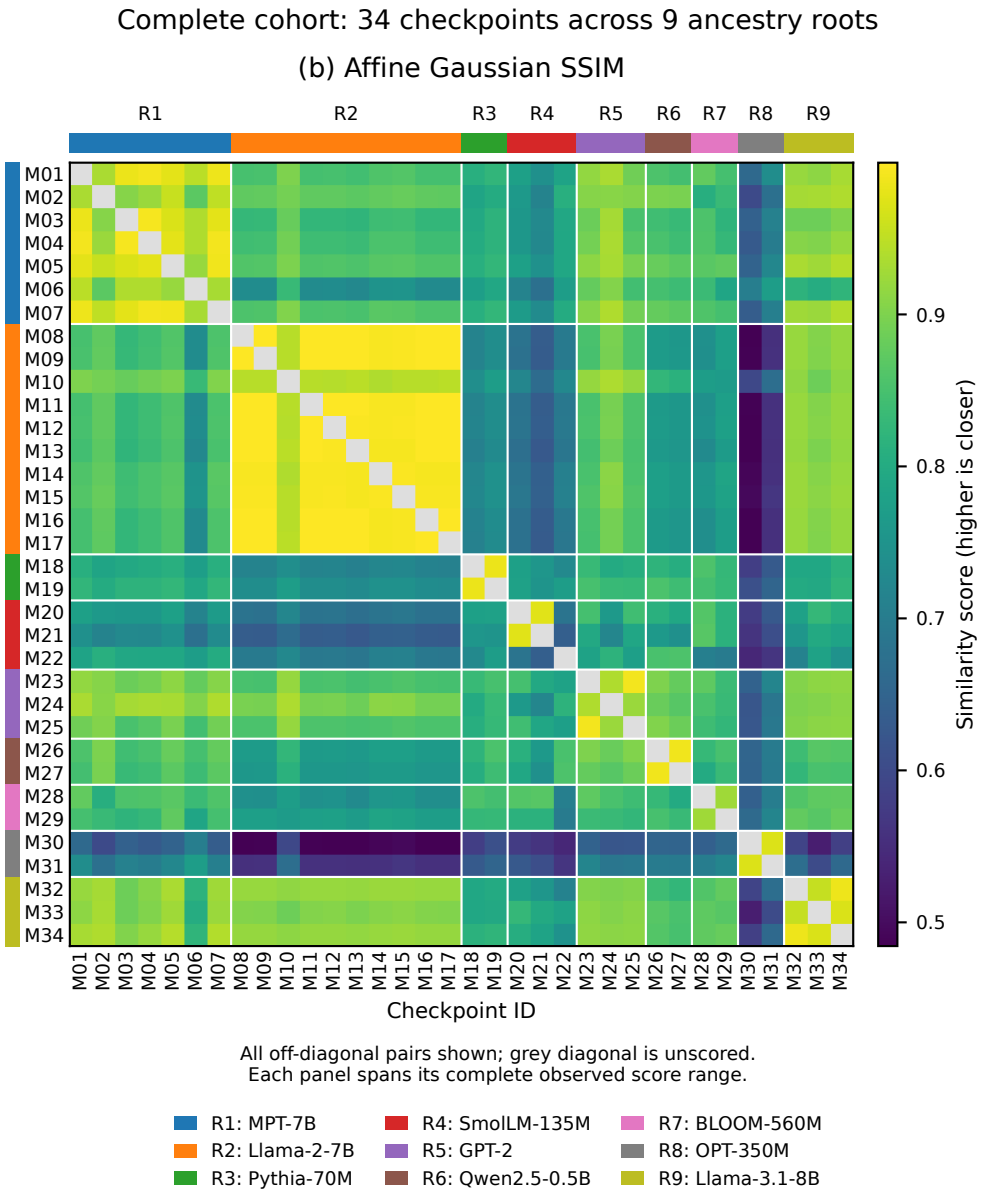

Figure 6: Complete-cohort SSIM scores using the same fingerprints and checkpoint ordering as Figure 5. Higher values indicate closer PAD fingerprints. White lines delimit the nine model families, and diagonal cells are unscored. Each heatmap uses its own complete observed score range.

A.3 CONTROLLED DETECTOR-TARGETED ADAPTATION

Executed configuration. The controlled study starts from Llama-2-7B and updates rank-8 LoRA adapters for 200 steps. Both conditions use BF16, AdamW with learning rate 2×10^{-5}, maximum sequence length 256, seed 42, and the same initialization, data, and minibatch order. The PAD-aware condition adds the detector-targeted objective described in the main text; the matched control performs utility-only adaptation.

Splits and evaluation. The 200 probe rows are split by source question into 120 training, 40 development, and 40 test rows, containing 107, 36, and 36 distinct source questions, respectively; variants of the same source question never cross partitions. The attack configuration is selected on the development split and then fixed, while all reported fingerprint scores use the 40 held-out test pairs. Utility data comprise 128 training, 32 development, and 64 test items derived from MMLU (Hendrycks et al., 2021). The fixed test diagnostic reports full-text PPL and a zero-shot multiple-choice check; it is not the complete official MMLU benchmark.

Table 7: Complete trajectory of the controlled PAD-aware attack and its matched utility-only control. PAD sim. reports the same similarity score used in the main text. PPL/base and MCQ use the same independent 64-item diagnostic.

Objective	Updates	PAD sim.	PPL/base	MCQ
Utility-only	0	0.9888	0.9997	24/64
Utility-only	50	0.9864	0.9452	23/64
Utility-only	100	0.9841	0.8789	23/64
Utility-only	200	0.9840	0.8657	28/64
PAD-aware	0	0.9888	0.9997	24/64
PAD-aware	50	0.9845	0.9716	24/64
PAD-aware	100	0.9773	0.9404	23/64
PAD-aware	200	0.9432	0.8924	27/64

This controlled result covers one checkpoint, one optimization seed, and one LoRA parameterization. No independently calibrated threshold is available for the 40-pair protocol, so the raw score trajectory is not an attack-success rate. The separate PAD-targeted stress test in Table 2 is retained as a recorded historical result and is not combined with this controlled trajectory.

B REASONS FOR LINEAR CKA

PAD requires an alignment operator $A(\cdot, \cdot)$ that (i) compares representations across *different feature dimensions* (e.g., different hidden sizes or layer widths), (ii) is robust to benign reparameterizations of the representation space (e.g., neuron permutations and uniform scaling), and (iii) remains computationally practical when repeatedly applied to many layer pairs. We instantiate A with **Centered Kernel Alignment (CKA)**(Kornblith et al., 2019), a normalized dependence measure built on the Hilbert–Schmidt Independence Criterion (HSIC)(Gretton et al., 2005). CKA has been shown to reliably compare neural representations even when their dimensionalities differ, which matches the cross-model comparison needs in PAD.

Definition. For matrices $U \in \mathbb{R}^{m \times p_1}$ and $V \in \mathbb{R}^{m \times p_2}$ computed on the same m aligned examples, CKA is defined as

$$\mathrm{CKA}(U, V) \;=\; \frac{\mathrm{HSIC}(U, V)}{\sqrt{\mathrm{HSIC}(U, U)\,\mathrm{HSIC}(V, V)}}, \tag{3}$$

where

$$\mathrm{HSIC}(U, V) \;=\; \frac{1}{(m-1)^2}\,\mathrm{tr}\big(K_U\, H\, K_V\, H\big). \tag{4}$$

Here $H = I - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$ is the centering matrix, and $K_U, K_V \in \mathbb{R}^{m \times m}$ are Gram matrices defined by a kernel k over rows (Zhang et al., 2025a):

$$(K_U)_{ij} = k(U_{i,:}, U_{j,:}), \qquad (K_V)_{ij} = k(V_{i,:}, V_{j,:}).$$

Robustness to representation reparameterization. A key motivation for CKA in our setting is its invariance to common column-wise transformations of representations. In particular, linear CKA is invariant to (i) orthogonal transformations, including permutations of representation dimensions, and (ii) uniform scaling (Kornblith et al., 2019). This property is desirable for PAD because models from the same lineage may differ by post-training changes (e.g., pruning) or implementation-level reparameterizations, while still preserving the underlying panic-pattern structure that PAD aims to capture.

Kernel selection: linear vs. RBF. CKA allows different kernels to define the Gram matrices. In our experiments, we considered both:

- **Linear kernel:** $K_U = UU^\top$ (and similarly $K_V = VV^\top$).
- **RBF kernel:** $k(u_i, u_j) = \exp\big(-\|u_i - u_j\|_2^2/(2\sigma^2)\big)$.

We evaluate kernel choice by computing PAD similarities between llama-2-7b and all its variants shown in Figure 2. Empirically, RBF yields similarities above 0.83 for all variants, while linear yields similarities above 0.85. The average similarity is 0.9141 for RBF and 0.9121 for linear, indicating that both kernels are suitable for instantiating the PAD alignment operator and achieve comparable separation behavior in practice. Given this similarity in effectiveness, we adopt **linear CKA** by default due to its substantially lower computational cost, which is critical for constructing layerwise alignment matrices over many layer pairs.

C Supplementary PAD Results

We report an additional PAD result using **Llama-2-7b** as the victim model to complement the main-text evaluation. Table 8 shows that PAD assigns high similarity to the in-family fine-tuned derivative (Column 1) while keeping low similarities for unrelated model families, providing an extra data point consistent with the main conclusions.

Table 8: Supplementary PAD results with victim model: **Llama-2-7b**. Cell colors indicate similarity: > 0.8, 0.5–0.8, < 0.5.Column 1 is an in-family fine-tuned derivative; others are unrelated families.

	Victim Model: Llama-2-7b						
Method	**Llama-2** chinese-llama-2-7b	**Llama-3** llama-3.1-8b-instruct	**Qwen** Qwen2-7b	**Mpt** mpt-7b	**Deepseek** deepseek-r1-7b	**GPT-OSS** gpt-oss-20b	**InternLM2** internlm2-7b
Logits	0.7001	0.0000	0.0900	0.9933	0.6567	0.0033	0.9999
REEF	0.9977	0.1942	0.1339	0.1709	0.0748	0.1244	0.1568
Adv-REEF	0.9970	0.1959	0.1294	0.1366	0.0554	0.1541	0.1498
PAD	0.9467	0.0000	0.005	0.0205	0.0000	0.001	0.007

D OQT: Questions, Estimator and Thresholds

Threat model and budget. The adversary controls the suspect endpoint and may wrap it with a system prompt, post-process its responses, change its decoding parameters or fine-tune it, but does not see the questions in advance. The verifier's budget is bounded and known in advance, since the reference fingerprints are computed once and reused.

Select questions and collect responses. We select a set Q of questions that admit a range of answers by sampling from an open-ended question set. Identifying such questions beforehand is difficult, and random sampling is adequate at this size; a pruning procedure that uses OQT itself as feedback can shorten the set further (Appendix F). For every $q \in Q$ we collect $\mu_f(q)$, and the fingerprint of f is the resulting set of per-question mean embeddings, which is stored and compared like any other fingerprint.

Elicitation conditions. The fingerprint is defined over a small family of *conditions*: the same questions under different system prompts, which decide how much of the answer the model has to choose for itself. A condition that pins the style down compresses the answer, and one that hands the style to the model asks it to commit to a manner that post-training, rather than the question, determines. What a condition fixes is shared by every model and carries no lineage information; what it leaves free is where the fingerprint lives. Each condition costs one pass over the question set and adds one vector per question to the fingerprint, so adding a condition is a pricing decision, and Appendix O reports the comparison.

Removing the consensus answer. Comparing $\mu_f(q)$ with $\mu_g(q)$ measures how alike two models answer, not whether they answer alike *for the same reason*: on any question with a popular answer, most of the similarity is that popular answer. The score therefore compares what is left after removing what the reference pool says about the question. Writing $\bar{\mu}(q)$ for the mean of $\mu_f(q)$ over the models the verifier has already fingerprinted,

$$\tilde{\mu}_f(q) \;=\; \mu_f(q) - \bar{\mu}(q), \qquad \mathrm{Sim}_q(f,g) \;=\; \cos\big(\tilde{\mu}_f(q), \tilde{\mu}_g(q)\big),$$

For the fixed-pool result we average the per-question scores over $q \in Q$. The line-held-out diagnostic applies a light lower-tail winsorization before the mean, and forms $\bar{\mu}$ from vendor lines other than the two being compared, so a pair is never measured against a consensus it contributed to. The pool is built once from models the verifier already holds—the suspect is never part of it—so removal costs no additional queries and makes the score a statement about the model rather than about the question. The aggregation is then a fixed choice, compared against the alternatives in Appendix H.

Sim_Q is continuous and relative, so a provenance decision requires a threshold, and the useful cut-off depends on which question is being asked (lineage verification is stricter than family classification), on the embedder and question set, and on how an auditor weighs a false accusation against a missed theft. We therefore make the boundary an explicit part of the method rather than an incidental constant:

1. calibrate on model pairs *disjoint* from the pairs used for evaluation, so the reported performance is not tuned on itself;
2. choose the threshold that maximizes F_1 on that calibration set;
3. report precision, recall, F_1 and the false-positive rate at a fixed true-positive rate on the held-out pairs, with the full precision–recall curve, so a deployment can move its operating point without re-deriving the score; and
4. report reference bands for our default setting, so raw scores can be read.

E OQT Implementation Details

Sample count. We use $r = 10$ as the default sample count of OQT, which is empirically sufficient under our settings. The count is what each model was asked for, not always what it returned: 31 of the 35 models delivered all 750 responses, three had smaller shortfalls, and DeepSeek-R1 returned 457 with as few as two samples on some questions. Table 19 lists the per-model counts. A fingerprint averages the samples a model actually returned, so incomplete collection makes that model's estimate noisier rather than incomparable. A split-half check in Appendix H compares two disjoint five-sample estimates for this cohort.

Probe conditions. The elicitation conditions of Appendix O are collected with three samples per question on the reduced pilot subset reported there, which keeps the comparison between conditions affordable; the main collection uses ten samples over all seventy-five. Responses are embedded with the same embedder and the same aggregation as the main experiment, so a condition differs from the main run only in the system prompt and in how much of the question set it covers.

Aggregation. The fixed-reference result uses the mean of the per-question similarities. The line-held-out diagnostic winsorizes the lower 2.5% before taking the mean; alternative aggregations are compared in Table 16.

Fitting GMM. We initialize the two-component GMM with K-means and use the **expectation–maximization (EM)** algorithm to fit this model. During EM, we constrain the parameters to $0.6 \leq \phi \leq 0.9, 0.01 \leq \sigma_1, \sigma_2 \leq 1$ to avoid degenerate solutions.

Aggregation selection. The mixture is not the estimator our score uses; it is the alternative we compare the winsorized mean against (Table 16). On the reported lineage task, the mixture aggregation reaches ROC-AUC 0.748, compared with 0.850 for the winsorized mean (Table 16).

Threshold calibration. The threshold is calibrated at the lineage level rather than on random pairs, and no evaluation pair contributes to it. The calibration set is the pairs whose two models both come from the Llama, Qwen or DeepSeek lines; the evaluation set is every pair that involves a model from any other line. The threshold is chosen to maximize F_1 on the calibration pairs, applied unchanged to the evaluation pairs, and the full precision–recall curve is reported so that other operating points can be read off (subsubsection 5.2.4). One asymmetry is structural and we state it rather than hide it in a row of zeros: every checkpoint-sharing cluster in this pool sits inside the calibration lines, so the evaluation set holds no pair of that kind, and the lineage threshold cannot be transfer-tested here. The transfer test is therefore run on the co-release task, which does have pairs on both sides of the split, and the lineage result is measured leave-one-model-out instead (subsubsection 5.2.1), which is the harder protocol: the held-out model does not appear on the reference side at all.

Score margins and confidence. Because the decision is taken on a continuous score, we report the distribution of scores for same-lineage and different-lineage pairs, the gap between them, and bootstrap confidence intervals for the scores of individual pairs, obtained by resampling the questions with replacement.

F Question Pruning

Method. The randomly sampled Q contains some useless questions that we can eliminate. To do that, we exploit the result of the OQT as feedback to determine the discriminative power of the questions.

First, we collect a set of reference models F containing representative models from different families, and calculate $\mathrm{Sim}_Q(f, g)$ for every two distinct models f, g in F. Then, for each question, we hypothetically remove it, calculate the new scores, and compare them with the original using the mean squared error (MSE). Finally, the question whose removal causes minimal MSE is removed, producing Q_1 that contains exactly one fewer question than Q. Repeating this greedy process, we can produce Q_i that contains i fewer questions than Q for any i.

These equations describe this step formally:

$$
\begin{aligned}
Q_0 &= Q \\
Q_i &= Q_{i-1} \backslash \{q_i\} \\
q_i &= \arg\min_{q \in Q_{i-1}} \mathrm{MSE}_i(q) \\
\mathrm{MSE}_i(q) &= \frac{1}{|F|\,(|F|-1)} \sum_{\substack{f,g \in F \\ f \neq g}} \mathrm{erf}_i(q, f, g)^2 \\
\mathrm{erf}_i(q, f, g) &= \mathrm{Sim}_{Q_{i-1} \backslash \{q\}}(f, g) - \mathrm{Sim}_Q(f, g)
\end{aligned}
$$

Afterwards, we choose one of these sets whose MSE is acceptable and stable as the pruned set.

Objectives. Two objectives are available. The *stability* objective removes the question whose removal perturbs the similarity matrix least (measured as MSE against the full question set), which keeps a pruned set that reproduces the full-set scores. The *margin* objective removes the question

Table 9: OQT on pruned question sets, evaluated on every candidate model, including those outside the pruning reference set.

Objective	Questions	F_1	Precision	Margin	Threshold
margin	55	0.362	0.309	+0.130	0.228
mse	55	0.361	0.314	+0.112	0.236

that leaves the largest gap between the median similarity of same-lineage pairs and that of different-lineage pairs, which keeps a pruned set that is discriminative rather than merely stable. A pruning rule that only minimises the MSE against the full set is not aligned with the decision the score is used for, since a small perturbation of an already-poorly-separated matrix is cheap in MSE terms; we therefore report both and use the margin objective for the pruned sets below.

Effectiveness We perform feedback pruning with one representative model per lineage line as the reference set, and evaluate the resulting question sets on *all* candidate models, including those not used as references in the pruning step. Table 9 reports the result.

Both objectives keep 55 of the 75 questions (27% fewer), and the pruned sets stay usable on the models the selection never saw: on all 35 evaluated models the stability objective reaches $F_1 = 0.361$ (precision 0.314, recall 0.423) and the margin objective $F_1 = 0.362$ (precision 0.309, recall 0.436), against $F_1 = 0.417$ at precision 0.326 and recall 0.577 for the full question set (Table 9). Pruning therefore reduces the query budget at the cost of both precision and recall in this cohort; it preserves a usable ordering rather than improving decision quality.

What pruning does not do is increase the separation. The margin objective turns the margin on the reference models it selects on from -0.01 to $+0.069$, so on those twelve models the selection does what it says; applied to all 35 models the same set leaves 0.13 and the stability set leaves 0.112, against 0.202 for the full question set. The questions it removes are the ones whose removal least perturbs a matrix that already ranks the lineages in the right order, so a pruned set preserves that ordering on a smaller budget rather than sharpening it. Figure 7 shows both halves of that statement: on the reference set the margin objective's line crosses zero while the stability objective's stays just below it, and on all 35 models the two sets land at 0.13 and 0.112, below the dashed full-question-set margin of 0.202.

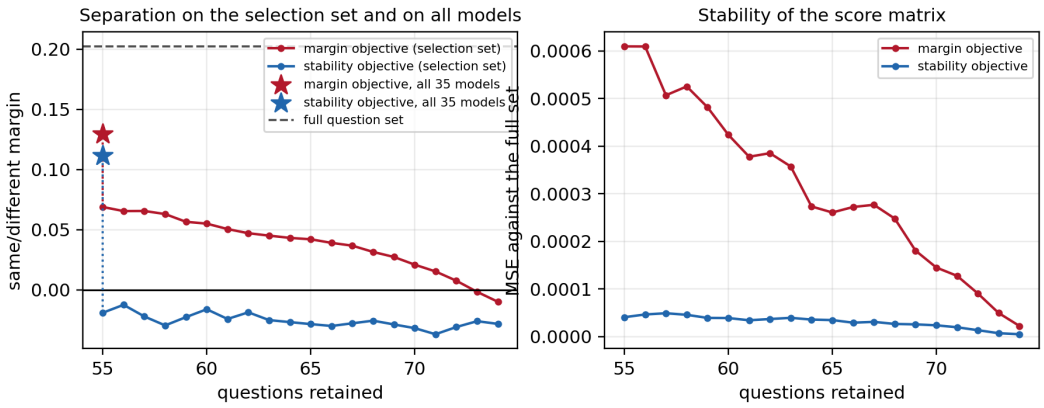

Figure 7: Question pruning as an optimisation. Left: the same-lineage versus different-lineage margin on the twelve reference models the selection was performed on (lines), and the same margin when the pruned set is applied to all 35 evaluated models (stars, joined to the end of their own line by a dotted segment: the two points describe the same pruned set, evaluated on the selection set and on every model). Right: the perturbation of the score matrix relative to the full question set. Both objectives improve the margin on the reference set while removing a quarter of the questions; on the full set the pruned matrices keep a smaller margin than the full question set, so what transfers is a usable ordering, not greater separation or decision quality.

Table 10: The main embedder against five alternatives, all recomputed from the same responses.

Embedder	Embedding size	ROC-AUC	Margin	Threshold	Rank corr. with main
`qwen3-embedding-8b`	4096	0.732	+0.202	0.346	–
`qwen3-embedding-4b`	2560	0.716	+0.193	0.368	0.988
`bge-large-en-v1.5`	1024	0.770	+0.193	0.333	0.942
`mistral-embed-2312`	1024	0.777	+0.206	0.305	0.935
`e5-large-v2`	1024	0.787	+0.191	0.289	0.929
`gte-large`	1024	0.791	+0.193	0.237	0.908

G OQT Results with Alternative Embedding Models

OQT is not invariant to the embedding model, so we repeat the full experiment with five further embedders of different sizes and training objectives: a smaller member of the main embedder's family (`qwen3-embedding-4b`, 2560 dimensions), a classical retrieval embedder (`bge-large-en-v1.5`, 1024), a multilingual embedder (`mistral-embed-2312`, 1024), a sentence-embedding baseline (`e5-large-v2`, 1024) and a general-purpose text embedder (`gte-large`, 1024). For each of them we re-embed every response, recompute the similarity matrix, recalibrate the threshold on the same calibration pairs and evaluate on the same held-out pairs. Table 10, included below, reports the result, and three things follow.

Every one of the six embedders separates the line task under the residual score, at ROC-AUC 0.716–0.791, so the score is reading a property of the responses rather than of a particular embedding space. The ranking of embedders is not the ranking of general retrieval quality: our main embedder is not the best of the six here (a 1024-dimensional retrieval embedder reaches 0.791), which is why we fix the embedder and calibrate the threshold per setting instead of choosing the embedder that looks best. And the embedder changes the scale more than the ordering: the Spearman correlation between the main embedder's per-pair scores and the others' is 0.908–0.988, while the thresholds that maximize F_1 move from 0.237 to 0.368. The absolute and residual scores both vary with the embedding model, as reported in the table above. Three of the alternatives (`bge`, `e5`, `gte`) accept only 512 input tokens, while a reasoning model's answer can be longer; for those we truncate each response to the embedder's limit before embedding, the standard treatment for a short-context retriever, which can only remove information and does not affect the main results.

H Effect of Consensus Removal

We compare absolute response similarity with the residual score on the same evaluated pairs. Both scores are computed from the same response embeddings.

Removing the consensus changes the separation on the evaluated pairs. On the lineage task ROC-AUC rises from 0.644 to 0.840, and to 0.849 when the pair's own vendor line is held out of the reference pool as well—the variant the rest of this section uses—with the margin going from $+0.018$ to $+0.315$; on the one-vendor-line set, from 0.567 to 0.722 (0.718 held out); and on documented-derivative pairs against every other pair, from 0.650 to 0.861 (0.866) (Table 11 names every row and column). These cells come from the stored-embedding comparison of this subsection rather than from the pipeline of subsubsection 5.2.1, and the two differ slightly because the pipeline winsorizes the per-question similarities: the lineage task reads 0.849 here and 0.850 there, the one-vendor-line task 0.718 here and 0.732 there.

In the reported controls, against a permutation null over the pair labels the score's 0.849 on the lineage task gives $p < 0.001$ (null mean 0.501, 95th percentile 0.637), while scoring pairs by equal residual magnitude reaches 0.465 and by equal response length 0.548, with a correlation of $+0.10$ to the score, so the signal lives in the *direction* of the residual rather than in its size.

Two disjoint five-sample estimates correlate at $\rho = 0.993$ and give ROC-AUC 0.723 and 0.727 on the relaxed task (0.855 and 0.857 on the derivative task); their average gives 0.725 on the relaxed task in this cohort.

Table 11: The same pairs scored three ways, all three computed from the stored embeddings by one script so that the only thing that changes across a row is the correction. *Absolute* is the mean question similarity; *Residual* removes the mean answer of the whole reference pool first; *Residual, line held out* removes the mean answer of the pool with the pair's own vendor line excluded, which is the variant the rest of this section reports; Appendix H states where the pipeline's own estimator differs from these cells and why. *Same* counts the positive pairs, and a row marked – holds pairs of a single class. *One vendor line* counts every pair whose two models share a vendor, *calibration lines* and *held-out lines* are the split of subsubsection 5.2.4, and *documented derivatives* keeps the positive class to pairs with a documented derivation and compares them against every other pair.

Pair set	Pairs	Same	Absolute	Residual	Residual, line held out
Checkpoint-sharing pairs	595	12	0.644	0.840	0.849
One vendor line	595	78	0.567	0.722	0.718
Calibration lines	210	72	0.565	0.743	0.697
Held-out lines	385	6	0.751	0.794	0.756
Documented derivatives	529	12	0.650	0.861	0.866

Table 12: Decision quality at the calibrated threshold, chosen on the calibration pairs and applied unchanged to the held-out pairs. The lineage row is marked n/a because every checkpoint-sharing pair in this pool lies inside the calibration lines.

Task	Threshold	F_1 on calibration pairs	Precision	Recall	F_1	Accuracy
strict	0.489	0.429		n/a		0.990
relaxed	0.223	0.586	0.057	0.667	0.105	0.823

I Threshold Calibration and Reference Bands

Calibration and transfer. The threshold is chosen to maximize F_1 on the pairs whose two models come from the Llama, Qwen or DeepSeek lines (210 pairs) and applied unchanged to every pair that involves a model from another line (385 pairs); Table 12 reports both sides. On the calibration pairs the procedure looks usable ($F_1 = 0.586$), which is the number a report would quote if it stopped there; Figure 8 shows that the trade-off has no plateau, so raising precision above chance costs most of the recall. On the held-out lines it degrades to precision 0.057 at recall 0.667: it finds two thirds of the co-release pairs it should find, and includes about sixteen unrelated pairs for each of them. A threshold that reaches a true-positive rate of 0.95 costs a false-positive rate of 0.923, so an auditor who wants to miss at most one co-release in twenty must accept that roughly nine in ten of their accusations are wrong. The lineage row of the table is marked n/a because every checkpoint-sharing cluster in this pool sits inside the calibration lines, so there is no pair of that kind to transfer to; the lineage measurement is the leave-one-model-out protocol of subsubsection 5.2.1, which is harder, since the held-out model does not appear on the reference side at all.

Reference bands. With the default setting, the calibrated threshold is 0.22; same-lineage pairs score between -0.28 and 0.64 (median 0.27) and different-lineage pairs between -0.41 and 0.67 (median 0.07). The two ranges overlap on [-0.28, 0.64]: 78% of the same-lineage pairs score above the different-lineage median and 14% of the different-lineage pairs score above the same-lineage median, so a band that excludes false positives also excludes positives.

When the two distributions are separated, a score at or above the threshold is evidence for the lineage hypothesis and a score well below it is evidence against. The overlap is the band in which an auditor should collect more evidence—the white-box fingerprint of subsection 4.1, when the model is available—rather than treat the score as a verdict. The bands move with the correction in the direction the account predicts, but they remain wide: a threshold that excludes the top 7% of the different-lineage pairs keeps 79% of the same-lineage ones. That is enough to rank a suspect against a pool of candidates (subsubsection 5.2.1) and not enough to pronounce on a single pair. Table 18 lists the different-lineage pairs that sit highest—the top one at 0.672, inside the same-lineage interquartile range—which is where any threshold has to be paid for.

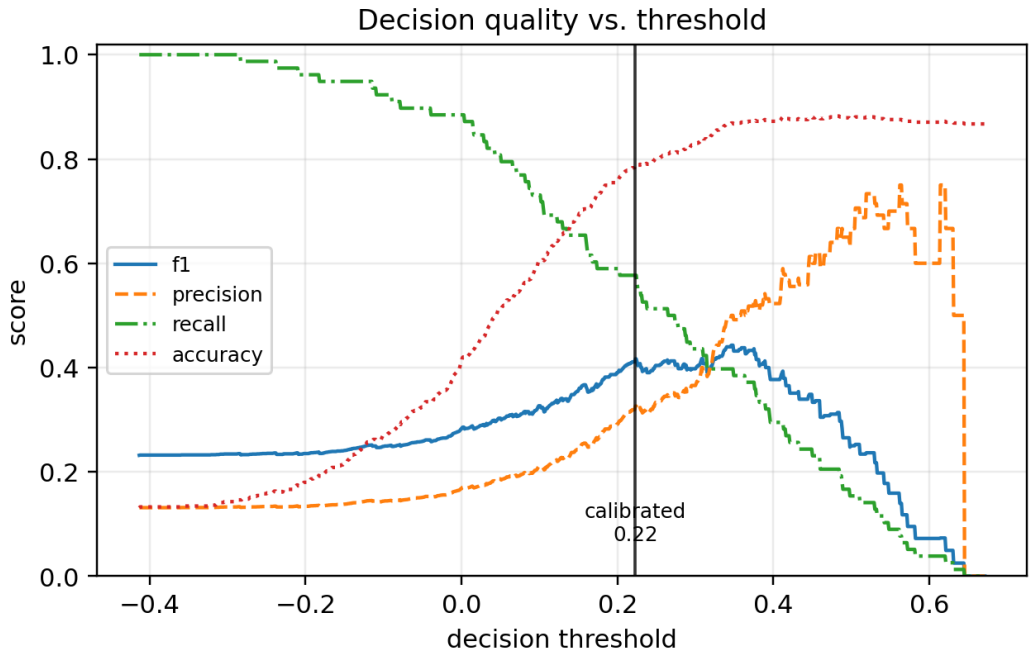

Figure 8: Precision, recall, F_1 and accuracy as a function of the threshold, with the calibrated value marked. There is no plateau: because the two score distributions overlap, every cut-off that raises precision above chance costs most of the recall.

Table 13: How many reference models the consensus needs. The score is recomputed end to end with the reference pool restricted to a random subset of the models outside the candidate pair's two vendor lines; entries are the mean and the standard deviation over 8 draws. *Checkpoint-sharing* is the strict task of subsubsection 5.2.1 and *same line* the relaxed one.

Reference pool	Checkpoint-sharing	Same line
2	0.688 (0.113)	0.612 (0.083)
4	0.739 (0.087)	0.667 (0.067)
8	0.790 (0.030)	0.695 (0.035)
16	0.815 (0.014)	0.711 (0.013)
all eligible	0.812 (0.000)	0.705 (0.000)

J Robustness: Reference Pool, Post-Processing and Imitation

Reference pool size. The score removes the mean answer of the verifier's reference pool, so the verifier also chooses how many models that pool needs. Table 13 recomputes the score end to end with the pool restricted to k models drawn from outside the candidate pair's two vendor lines, over eight draws each. A few reference models already recover most of the separation: the checkpoint-sharing ROC-AUC is 0.688 at $k = 2$, 0.739 at $k = 4$, 0.790 at $k = 8$ and 0.815 at $k = 16$, against 0.812 for the whole eligible pool, and the co-release task follows the same curve (0.612 at $k = 2$ rising to 0.711 at $k = 16$, against 0.705 for the whole pool). What the pool buys beyond $k = 16$ is stability rather than separation, the spread over draws falling from 0.113 at $k = 2$ to 0.014 at $k = 16$. A verifier does not need a large reference population to use the score, which matters because every model in it must be fingerprinted first.

Post-processing. The cheapest attack is not imitation but post-processing: wrap the endpoint, truncate the answers, or rewrite them with another model. Table 14 applies each of these to the stored responses and rescores the pool end to end, on the models that carry a checkpoint-sharing relation. Truncating each answer to its first two sentences and prepending a politeness prefix leave

Table 14: Survival of the score under post-processing of the answers. Every row is scored with the same estimator on the same 25 questions and 17 models, so only the text differs; *change* is relative to the unperturbed row. The last row rewrites each answer with nova-lite-v1, from a vendor absent from the pool.

Post-processing	Lineage ROC-AUC	Change
Unperturbed	0.838	—
Truncated to the first two sentences	0.886	+0.048
Politeness prefix prepended	0.845	+0.007
Lowercased, punctuation stripped	0.807	-0.031
Rewritten by an unrelated model	0.777	-0.061

the ranking intact (0.886 and 0.845 against 0.838); stripping case and punctuation costs 0.031; and a full rewrite by an unrelated model, Amazon Nova Lite, costs 0.061 and still leaves the score at 0.777, far above chance. What survives a rewrite is the component a fingerprint should rely on: nothing here depends on surface form, and the strongest of the cheap attacks degrades the decision rather than destroying it.

Adaptive attack: in-context imitation. The cheapest attack a black-box adversary can mount is behavioural imitation: show the suspect examples of the victim's answers and ask it to answer in that style. We collect the imitation under $k \in \{0, 15, 30\}$ question/answer pairs drawn from the victim's own responses and prepended to the imitator's system prompt (Appendix Q), and fingerprint it like any other model; the imitator is never a fine-tune of the victim, because a fine-tune is the relation the fingerprint is meant to find. Table 15 reports six victim/imitator pairs, drawn from two different vendor lines each, over eighteen conditions, and two facts hold in all of them. Imitation moves towards the victim as examples are added: the similarity to the victim rises monotonically with k in every pair, from -0.130 at $k = 0$ in one pair to 0.635 at $k = 30$ in another. With no leaked questions the victim is never the imitation's closest model—its rank in the pool runs from 4 to 25 of 35—although two pairs already exceed the relaxed threshold. At $k = 30$, the victim becomes the closest model in two of the six pairs, and four of six pairs exceed the relaxed threshold of subsubsection 5.2.4; two of those threshold crossings are newly induced by the imitation. An adversary who holds the verifier's questions and enough of the victim's answers can therefore compromise retrieval and can create additional threshold false positives; what limits the attack is the threat model, in which the questions are not public, and the collection the adversary must pay for. Fine-tuning, a longer imitation prompt and a threshold chosen for the lineage task are untested here.

K Collection Cost

We record the tokens, cost and wall-clock time of every request, so the cost of fingerprinting a model is measured rather than estimated. Table 19 summarizes the collection for the most expensive models; the table reports the most expensive cases together with their request and token counts.

Where the cost goes. The dominant term is generated tokens, and its size is set by whether the model deliberates before answering rather than by the question set: each question costs one request per sample—750 requests at our 75×10 setting—but the response length differs by more than an order of magnitude across models. Two consequences follow. First, the number of questions can be reduced with the pruning procedure of Appendix F at a roughly linear saving. Second, fingerprinting reasoning models is the expensive case, and the same mean-of-samples estimator can be applied to truncated deliberation traces, which we leave to future work. The comparison and aggregation steps are negligible: they are matrix operations over a $35 \times 35 \times 75$ tensor.

Table 15: In-context imitation: k question/answer pairs from the victim's own responses are prepended to the imitator's prompt. *To victim* and *to own base* are the imitation's scores against the victim and against the imitator's own base model; *victim rank* is the victim's position among the 35 candidates under the same protocol; *above thr.* counts the candidates at or above the calibrated threshold, and *attributed* says whether the imitation's similarity to the victim reaches it.

Leaked pairs k	To victim	Victim rank	Nearest model	To own base	Above thr.	Attributed
0	0.567	4/35	phi-4 (0.69)	0.695	7	yes
15	0.626	2/35	phi-4 (0.64)	0.638	8	yes
30	0.635	1/35	phi-4 (0.59)	0.592	9	yes
0	-0.107	22/35	deepseek-r1 (0.68)	0.626	12	no
15	-0.040	19/35	deepseek-r1 (0.72)	0.637	12	no
30	0.011	19/35	deepseek-r1 (0.70)	0.624	11	no
0	-0.037	22/35	aion-rp-llama-3.1-8b (0.63)	-0.270	7	no
15	0.047	13/35	phi-4 (0.47)	-0.247	7	no
30	0.200	9/35	phi-4 (0.35)	-0.196	6	no
0	0.045	17/35	aion-rp-llama-3.1-8b (0.60)	-0.002	8	no
15	0.233	8/35	llama-4-scout (0.51)	0.114	7	yes
30	0.281	7/35	llama-4-scout (0.49)	0.076	6	yes
0	-0.130	25/35	deepseek-r1 (0.62)	-0.109	11	no
15	0.309	2/35	deepseek-v4-flash (0.32)	0.291	6	yes
30	0.500	1/35	deepseek-v4-flash (0.30)	0.259	8	yes
0	0.265	10/35	glm-5.3-flash (0.66)	0.661	12	yes
15	0.401	4/35	glm-5.3-flash (0.63)	0.632	11	yes
30	0.420	3/35	glm-5.3-flash (0.56)	0.558	10	yes

Consensus removed	Aggregation	Lineage ROC-AUC	F_1	Prec.	Rec.	rank-1	Mean rank	Co-rel. ROC-AUC
none	mean	0.644	0.185	0.119	0.417	3/12	10.0	0.549
per-question mean	mean	0.840	0.286	0.250	0.333	3/12	5.7	0.689
leave-one-out	mean	0.850	0.375	0.300	0.500	4/12	5.8	0.698
leave-one-out	median	0.846	0.414	0.353	0.500	3/12	5.9	0.710
leave-one-out	trimmed mean	0.853	0.370	0.333	0.417	3/12	5.8	0.699
leave-one-out	winsorized mean	0.850	0.375	0.300	0.500	4/12	5.8	0.697
leave-one-out	mixture	0.748	0.250	0.250	0.250	2/12	9.3	0.633
leave-one-out	rank	0.774	0.182	0.143	0.250	3/12	9.5	0.509

Table 16: Which consensus is removed before comparing, and how the per-question similarities are aggregated, on the lineage task. The last two columns are the ranking view and the co-release task on the same pairs.

L OQT Scoring and Model Registry

Table 17: Candidate models used in the OQT experiments, in the order they are numbered throughout the section. "Base and modification" states the documented relation used as ground truth: two models are a *lineage* pair when they share a base checkpoint (a fine-tune, a merge, a pruning, or a second instruction/reasoning branch of one release), a *co-release* pair when they belong to one vendor line without sharing a checkpoint (successive or parallel releases at a different scale or architecture), and *unrelated* otherwise. "CoT" marks the models that emit a deliberation trace. The two relations are evaluated separately.

#	Model	Line	Base and modification	CoT
1	llama-3.1-8b-instruct	llama3	Meta Llama-3.1-8B; instruct release	
2	aion-rp-llama-3.1-8b	llama3	Meta Llama-3.1-8B; fine-tune (roleplay)	
3	llama-3.1-70b-instruct	llama3	Meta Llama-3.1-70B; instruct release	
4	hermes-3-llama-3.1-70b	llama3	Meta Llama-3.1-70B; fine-tune (Hermes 3)	
5	l3.1-euryale-70b	llama3	Meta Llama-3.1-70B; fine-tune (roleplay)	
6	llama-3.2-1b-instruct	llama3	Meta Llama-3.1-8B; instruct release (pruned)	
7	llama-3.2-3b-instruct	llama3	Meta Llama-3.1-8B; instruct release (pruned)	
8	llama-4-scout	llama3	Meta Llama-4; MoE release (16 experts)	
9	llama-4-maverick	llama3	Meta Llama-4; MoE release (128 experts)	
10	qwen3-30b-a3b-instruct-2507	qwen	Qwen3-30B-A3B; instruction-tuned branch	
11	qwen3-30b-a3b-thinking-2507	qwen	Qwen3-30B-A3B; reasoning branch	✓
12	qwen3.8-27b	qwen	Qwen3.8; dense release	
13	qwen3.8-flash	qwen	Qwen3.8; efficiency release	
14	deepseek-r1	deepseek	DeepSeek-V3; RL reasoning post-training (from V3 base)	✓
15	deepseek-chat	deepseek	DeepSeek-V3; chat release	
16	deepseek-v4-flash	deepseek	DeepSeek-V4; efficiency release	
17	gemma-4-31b-it	gemma	Gemma-4; dense release	
18	gemma-4-26b-a4b-it	gemma	Gemma-4; MoE release	
19	glm-5.3-flash	glm	GLM-5.3; efficiency release	
20	nemotron-3-super-120b-a12b	nemotron3	Nemotron-3; 120B-A12B release	✓
21	nemotron-3-nano-30b-a3b	nemotron3	Nemotron-3; 30B-A3B release	
22	gpt-5.6-luna	gpt	GPT-5.6; efficiency tier	
23	gpt-5.4-mini	gpt	GPT-5.4; efficiency tier	
24	gpt-oss-120b	gpt-oss	GPT-OSS; open-weight release	✓
25	gemini-3.8-flash	gemini	Gemini-3.8; Flash tier	
26	muse-glimmer-30b	muse	Muse; efficiency release	
27	qwen3-next-80b-a3b-instruct	qwen	Qwen3-Next-80B-A3B; instruct release	
28	qwen3-next-80b-a3b-thinking	qwen	Qwen3-Next-80B-A3B; reasoning branch	✓
29	qwen3-32b	qwen	Qwen3-32B; instruct release (dense)	
30	llama-3.3-70b-instruct	llama3	Meta Llama-3.3-70B; instruct release	
31	deepseek-v3.2-exp	deepseek	DeepSeek-V3.2; instruct release	
32	gemma-3-27b-it	gemma	Gemma-3-27B; instruct release (previous generation)	
33	glm-4.6	glm	GLM-4.6; instruct release (previous generation)	
34	mistral-small-3.2-24b-instruct	mistral	Mistral-Small-3.2-24B; instruct release	
35	phi-4	phi	Phi-4; instruct release	

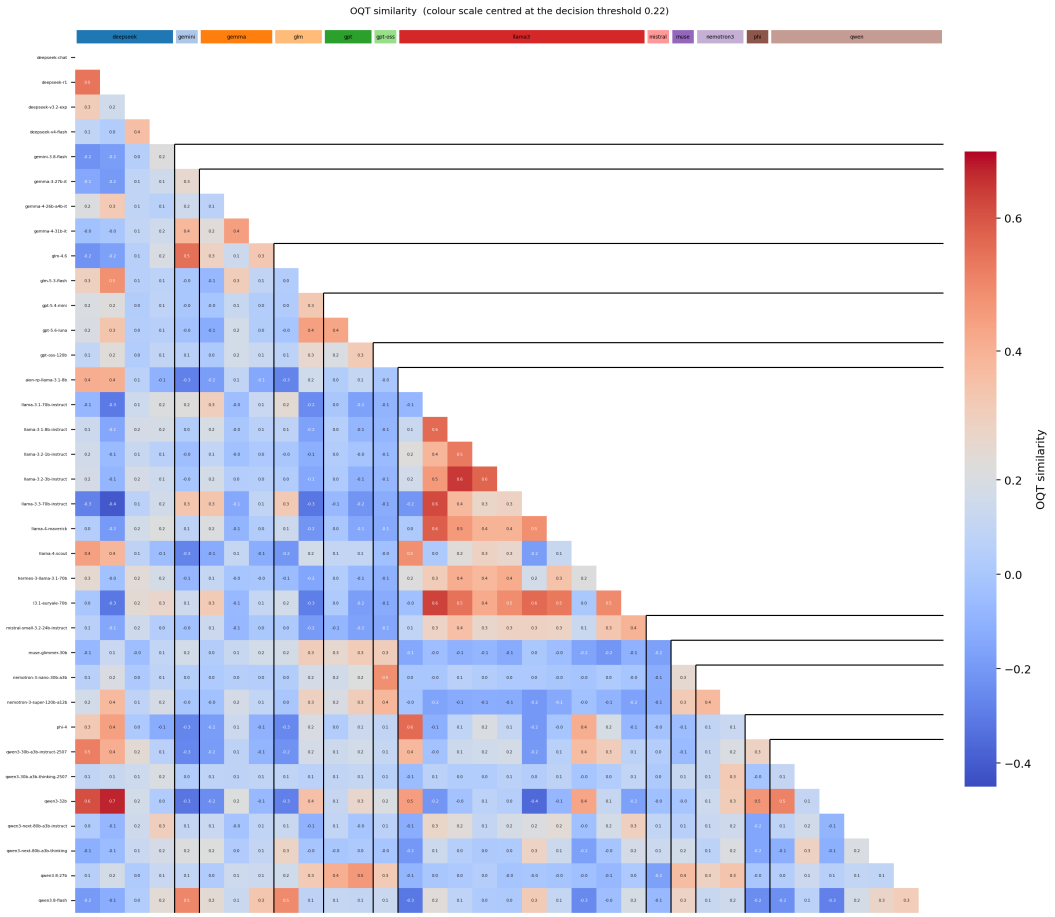

Figure 9: The full 35-model score matrix, all models of Table 17, drawn as one triangle with one-decimal cell values.

Table 18: The most confusable different-lineage pairs: these are the pairs that constrain the false-positive rate of any threshold.

Model f	Model g	Score	[95% CI]
deepseek-r1	qwen3-32b	0.672	[0.343, 0.701]
deepseek-chat	qwen3-32b	0.613	[0.205, 0.655]
aion-rp-llama-3.1-8b	phi-4	0.562	[0.315, 0.598]
gemini-3.8-flash	glm-4.6	0.548	[-0.141, 0.595]
qwen3-30b-a3b-instruct-2507	deepseek-chat	0.519	[-0.438, 0.578]
qwen3.8-flash	gemini-3.8-flash	0.504	[-0.153, 0.554]
qwen3.8-flash	glm-4.6	0.499	[-0.054, 0.544]
nemotron-3-nano-30b-a3b	gpt-oss-120b	0.487	[0.075, 0.506]
deepseek-r1	glm-5.3-flash	0.482	[0.154, 0.510]
qwen3-32b	phi-4	0.472	[0.192, 0.498]
qwen3.8-27b	gpt-5.6-luna	0.465	[-0.124, 0.510]
aion-rp-llama-3.1-8b	qwen3-32b	0.454	[0.057, 0.494]

M Additional OQT Detail

Per-pair detail behind the figures and tables of subsubsection 5.2.1: the most confusable different-lineage pairs, which constrain any threshold, the per-model collection cost, the scores of every pair in sorted order (Figure 10) and the ROC curves of the baselines (Figure 4).

Table 19: Collection cost per model, most expensive first. “Stored responses” is the fingerprint that was kept.

Model	Stored responses	Requests	Output tokens	Cost (USD)
qwen3-30b-a3b-thinking-2507	750	894	538K	1.30
glm-4.6	750	750	614K	1.17
deepseek-r1	457	906	443K	1.13
gemini-3.8-flash	746	752	254K	0.97
qwen3.8-27b	747	849	406K	0.94
qwen3-next-80b-a3b-thinking	728	750	610K	0.74
glm-5.3-flash	750	1284	873K	0.45
aion-rp-llama-3.1-8b	750	916	260K	0.45
muse-glimmer-30b	750	750	308K	0.40
gpt-5.4-mini	750	779	76K	0.36
qwen3.8-flash	750	849	547K	0.27
llama-3.1-70b-instruct	750	1634	317K	0.20

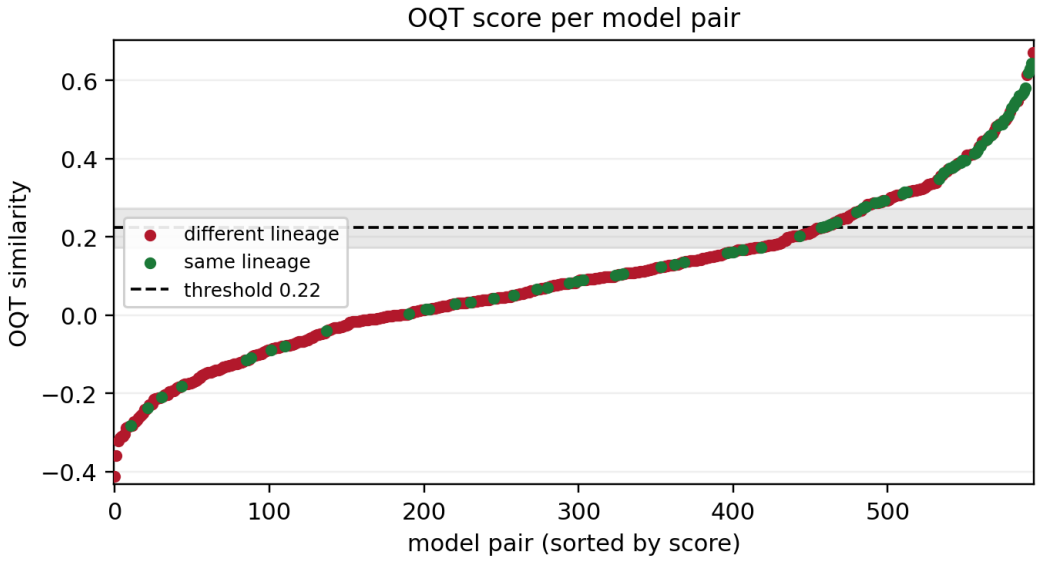

Figure 10: Every evaluated model pair, sorted by score and coloured by the lineage ground truth, with the calibrated threshold drawn in. The within-lineage pairs are concentrated at the top of the ordering rather than interleaved with the rest, and the pairs that remain inside the bulk of the distribution are the ones whose post-training moved their behaviour.

N BLACK-BOX BASELINES

OQT is compared against the direct comparison of responses and against the black-box baselines this problem invites. The direct comparison and our own learned baselines consume exactly the responses OQT consumes, so their query and token budgets are identical by construction. The two published methods do not, and we do not pretend otherwise: the hypothesis-testing tester is given our responses because that is what a text-only verifier can be given, while LLMmap is run as its authors released it, with its own queries and its own model, so its budget is *lower* than OQT's rather than higher. Each entry below names its source, states what we changed to fit a text-only interface, and quotes the number the released code computes.

Direct comparison of responses. The same similarity without the consensus removal, i.e. the mean of the question-level similarities. This is the ablation our submitted version reported and the aggregation a reader would reach for first; it reaches ROC-AUC 0.567, which is what the homogeneity of open-ended answers makes of a direct comparison (Jiang et al., 2025).

MMD two-sample test. For each question we compute the unbiased maximum mean discrepancy between the two response sets in embedding space with an RBF kernel (median-heuristic bandwidth), clip the estimate at zero, average over questions and map it to a similarity score; the test itself is the one of Gretton et al. (2012). It reaches ROC-AUC 0.556.

Hypothesis-testing provenance tester (Nikolic et al., 2025b). No implementation of this tester was released with the paper, so we reproduce the algorithm as published: the similarity of a candidate pair is the fraction of prompts on which the two models agree, each agreement is compared with the agreement of every control model by a two-proportion z test, and provenance is declared only when every null is rejected under a Holm–Bonferroni correction at $\alpha = 0.05$. Two substitutions are forced by our interface: the unit of agreement is the first word of the response (the modal word over our ten samples, standing in for their temperature-zero next token, since a text-only verifier sees no tokenizer), and the prompts are the same 75 open-ended questions rather than randomly sampled low-entropy-avoiding prompts. Their control set is the choice their paper shows to matter most, so we run both readings and report the more favourable one: models from third vendor lines only (1 of 595 pairs declared), and every model other than the pair (0 declared). Used as published, as a decision rather than a ranking, the tester reaches precision 1.00 and recall 0.013 against the relaxed one-vendor-line positive class. The reason it fires so rarely here is power: their design assumes on the order of a thousand prompts, and at 75 prompts the weakest of their comparisons needs an agreement gap of about 0.24 to reach significance after the correction, while the median agreement over our pool is 0.40 and the largest is 0.68—their own documented failure mode, a control set that contains models close to the parent. The number in Table 4 is the weakest evidence over controls (the minimum z), which is the ranking their statistic induces.

LLMmap (Pasquini et al., 2025). We do not reimplement this baseline: we call the authors' released implementation of LLMmap end to end against our endpoints—their eight queries, their prompt configurations, their template builder and their open-set inference model—and substitute only the backend that talks to a model, because none of their three backends reaches the endpoints we evaluate. Each model's template is built from 8 prompt configurations and it is scored on 3 held-out ones, 105 scored responses in total; their own default is far larger (100 configurations), which our API budget does not allow, and we say so rather than presenting the reduced run as theirs. Even so the comparison is not to our advantage: their protocol spends 88 short completions per model against the 750 responses OQT collects, so the budget it is given is smaller than ours rather than larger, and the accounting in Table 19 does not include it. Their model identifies the right model first for 55% of the scored responses and within the top three for 77% over the 35 evaluated models, which is their closed-set protocol restricted to our pool. Their headline result is higher—over 95% on 42 versions with eight queries—and the difference is the questions, not the classifier: their queries are built to elicit differences, while ours are open-ended questions whose answers models share, which is the premise of this paper. For the provenance task we score a pair

Elicitation	Lineage ROC-AUC	F_1	rank-1	Mean rank	Co-rel. ROC-AUC
brief (pilot subset)	0.693	0.600	4/11	3.5	0.680
long-form	0.673	0.375	1/11	5.3	0.620
stylised voice	0.779	0.452	4/11	3.8	0.759
brief, consensus removed	0.742	0.600	4/11	3.5	0.741
long-form, consensus removed	0.821	0.485	2/11	3.9	0.886
stylised voice, consensus removed	0.872	0.583	4/11	2.8	0.882
brief minus long-form	0.589	0.417	2/11	4.6	0.550
brief minus stylised voice	0.498	0.286	1/11	5.7	0.387

Table 20: Elicitation conditions on the pilot set (9 questions, 11 checkpoint-sharing pairs). Each condition is the same question set under a different system prompt: a brief answer, a long-form answer, and an answer in a stylised voice. The last two rows compare the brief condition with the other condition; the sign convention follows the row labels. After consensus removal, the long-form and stylised conditions both exceed the corrected brief condition, and the stylised condition is strongest; without that correction, long-form is slightly weaker than brief.

by the distance from the suspect to the closest member of the candidate parent's vendor line, taken in the direction most favourable to the baseline; that is the row in Table 4 (ROC-AUC 0.855).

Supervised pair classifier. Our own construction, and the learned reference that decides the relation from the responses alone: it uses per-question similarity, length and lexical features with a linear classifier under leave-one-lineage-out cross-validation, so no evaluated pair shares a lineage with its training pairs. It reaches ROC-AUC 0.657.

All methods, including OQT, are compared by ROC-AUC over all model pairs and by the best F_1 reachable by any threshold, together with the threshold that achieves it (subsubsection 5.2.2); the pair population is the same-line-against-different-line task of subsection 4.2, and a higher budget can only help a baseline, so the comparison is not rigged in OQT's favour. Two of the baselines are supervised: the pair classifier is trained on labels for the pair relation, and LLMmap is trained to name the models themselves, so it has a learner fitted to the very models under study. The ranking we care about is therefore not "which baseline is best" but whether removing the consensus answer is worth more than any of them on the same evidence, and where it is not, we say so. Where a baseline's authors report an operating point of their own we say so rather than mapping it onto our table: the hypothesis-testing tester is published as a decision at a controlled error rate, not as a similarity score, and its native operating point is what its paragraph above reports.

O Elicitation Conditions

The score reads the residual a model leaves once the consensus answer of the reference pool is removed, so one way to strengthen it is to ask the model for more of what the question does not determine. We collect the same questions under three system prompts—the brief instruction used elsewhere in the paper, a long-form instruction, and a stylised voice that the model has to sustain—on the models that carry the checkpoint-sharing relations, score every condition on its own, and score the difference between conditions, which cancels whatever two conditions share. Table 20 reports each condition, and three things follow.

In this pilot, the stylised condition separates the evaluated relations better than the brief and long-form conditions. The effect is not an accident of which questions the conditions share: a paired bootstrap puts the advantage of the stylised condition at $+0.152 \pm 0.050$ ROC-AUC on the lineage task and $+0.214 \pm 0.051$ on the line task, higher in every one of four hundred resamples. And the between-condition difference is not itself the stronger score—both difference rows stay below the stylised score alone—so the gain comes from what is measured rather than from the comparison. The stylised condition pays for itself in tokens like everything else here: its answers are two to four times longer than brief ones, so it is used where the fingerprint is calibrated rather than as a

default for every query. The comparison uses the same model subset and question subset across all conditions.

P AGGREGATION

Two aggregations are compared throughout. The *average* aggregation is the mean of the winsorized question-level similarities, and is the estimator the paper uses. The *mixture* aggregation fits a two-component Gaussian mixture to the same scores and reports the mean of the lower component, on the assumption that the questions split into one group that is informative about the pair and one that is not. That assumption is what it fails to deliver here. The mixture is a margin-sharpening device rather than a signal-creating one: it lowers the score of every pair, which helps only if same-lineage pairs are lowered less than different-lineage ones. On this model set the questions do not split that way, the criterion keeps selecting the mixture for most pairs, and the estimate it produces separates the classes less well than the plain average (Table 16). The group the mixture discards is the one that carries the pair's agreement with the rest of the pool, which is what removing the consensus has already accounted for, so on the residual score it is the discriminating questions rather than the easy ones that carry the signal. We therefore report the average as the estimator and the mixture as the ablation it is, and both aggregations are evaluated on the same question-level scores.

Q THE IN-CONTEXT IMITATION ATTACK

The attack of subsubsection 5.2.3 hands the adversary the resource a black-box provenance test is most exposed to—the victim's own answers—and asks whether showing them to another model makes that model look like the victim. For each pair we sample k questions from the victim's question set with a fixed seed, take the victim's first collected answer to each of them, and prepend the resulting exchanges to the imitator's system prompt:

> You are a model that imitates the assistant named *name*. Study the following example exchanges and answer every later question exactly in that assistant's style, tone and level of detail.
>
> Q: *question*
>
> A: *answer*
>
> ...

At $k = 0$ the system prompt is the neutral instruction used for the main collection, so that condition is the imitator's own fingerprint rather than an attack. We use $k \in \{0, 15, 30\}$ with four samples per question over the first thirty questions of the set, and every imitation is scored with the deployed pipeline—the same leave-one-line-out reference pool and the same calibrated threshold—so a condition counts as a leak only if the fingerprint would attribute the imitation to the victim's vendor line. Six pairs are run, each drawn from two different lines (Table 15). The imitator is a model of the pool rather than a fine-tune of the victim, because a fine-tune is already the relation the fingerprint is meant to find, and the question worth asking is whether imitation alone reaches it.

R SUPPLEMENTARY PRELIMINARY AND AGGREGATION RESULTS

Table 21: REEF similarity on clean vs. adversarial inputs (Llama-2-7B as reference). The first two models (vicuna-7b-v1.5, chinese-llama-2-7b) are in-family derivatives of Llama-2-7B, while the last one (Qwen2-7b) is from a different family. Cell colors indicate similarity: > 0.8, 0.5–0.8, < 0.5.

Data Type	vicuna-7b-v1.5	chinese-llama-2-7b	Qwen2-7b
Clean	0.9991	0.9977	0.1339
Adversarial	0.9993	0.9970	0.1294

Table 22: REEF similarity on clean vs. adversarial inputs (Mpt-7B as reference). All compared models (Qwen2-7b, Internlm2-7b, Baichuan2-7b) are from different families than Mpt-7B. Cell colors indicate similarity: > 0.8, 0.5–0.8, < 0.5.

Data Type	Qwen2-7b	Internlm2-7b	Baichuan2-7b
Clean	0.7789	0.8385	0.8568
Adversarial	0.6678	0.6762	0.6881

S ALGORITHM DETAILS

Algorithm 1 Panic Alignment Difference (PAD)

Input: Models f, g
  Aligned pairs $\{(x_i^{(c)}, x_i^{(a)})\}_{i=1}^m$
  Alignment operator A (linear CKA)
Output: PAD similarity score $S(d_\Delta(f, g))$
Let L_f and L_g be the number of layers of models f and g.
{Extract layerwise activations}
for $h \in \{f, g\}$ **do**
  Let L_h be the number of layers of h.
  for $l = 1$ to L_h **do**
    Form $X_{h,l}^{(c)}$ from $\{\phi_{h,l}(x_i^{(c)})\}_{i=1}^m$
    Form $X_{h,l}^{(a)}$ from $\{\phi_{h,l}(x_i^{(a)})\}_{i=1}^m$
  end for
end for
for $h \in \{f, g\}$ **do**
  Let L_h be the number of layers of h.
  for $l = 1$ to L_h **do**
    for $l' = 1$ to L_h **do**
      $S_h^{(c,c)}[l, l'] \leftarrow A\big(X_{h,l}^{(c)}, X_{h,l'}^{(c)}\big)$
      $S_h^{(a,c)}[l, l'] \leftarrow A\big(X_{h,l}^{(a)}, X_{h,l'}^{(c)}\big)$
      $\Delta_h[l, l'] \leftarrow S_h^{(c,c)}[l, l'] - S_h^{(a,c)}[l, l']$
    end for
  end for
end for
$d \leftarrow \mathrm{FrobeniusDistance}(\Delta_f, \Delta_g)$
$s \leftarrow \mathrm{Score}(d)$
return s

Algorithm 2 Open-ended Question Test (OQT)

Input: Models f, g; embedding model e; question set Q; sample count r
  Reference pool R; aggregation rule G
  Threshold τ (calibrated once on reference pairs, subsubsection 5.2.4)
Output: Score $\mathrm{Sim}_Q(f, g)$ and the decision $[\mathrm{Sim}_Q(f, g) \geq \tau]$
{Collect response embeddings}
for each model $h \in \{f, g\} \cup R$ **do**
  for each question $q \in Q$ **do**
    Initialize $A_h[q] \leftarrow \emptyset$
    for $i = 1$ to r **do**
      $s \leftarrow$ sample_response(h, q)
      Add $e(s)$ to $A_h[q]$
    end for
    $\mu_h[q] \leftarrow$ mean$(A_h[q])$
  end for
end for
{Remove the consensus answer of the reference pool}
for each $q \in Q$ **do**
  $\bar{\mu}[q] \leftarrow \mathrm{mean}_{h \in R}\, \mu_h[q] \{fixedreferencepool\}$
  $\tilde{\mu}_f[q] \leftarrow \mu_f[q] - \bar{\mu}[q]$
  $\tilde{\mu}_g[q] \leftarrow \mu_g[q] - \bar{\mu}[q]$
end for
{Compare the residuals and decide}
for each $q \in Q$ **do**
  $\mathrm{sims}[q] \leftarrow$ cosine_sim$(\tilde{\mu}_f[q], \tilde{\mu}_g[q])$
end for
$\mathrm{Sim}_Q(f, g) \leftarrow G(\mathrm{sims})$
return $\mathrm{Sim}_Q(f, g),\ [\mathrm{Sim}_Q(f, g) \geq \tau]$