RETROSPECTIVE DISTILLATION ATTRIBUTION VIA NORMALIZED RESPONSE SIMILARITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Model distillation transfers capabilities through supervised fine-tuning (SFT) on teacher responses, often collected from commercial APIs, raising questions of model provenance. Existing distillation attribution methods have been largely evaluated on students immediately after the SFT step. However, a distilled model may undergo further SFT, preference optimization, or reinforcement learning before release, while an auditor may lack access to the pre-distillation checkpoint required by reference-based attribution. To close this gap, we propose SCOUT, an output-only method that aggregates recurring *syntactic patterns* into candidate profiles, filters low-contrast patterns, and calibrates student–candidate distances against inter-candidate distances. SCOUT supports attribution and abstention using only current texts, without model weights, token likelihoods, or historical checkpoints. Auditing publicly released descendants of distilled models spanning diverse post-training objectives, SCOUT consistently identifies the distillation source. Furthermore, tracing teacher-associated *syntactic signatures* along training trajectories reveals that they emerge during distillation and persist through subsequent preference optimization and reinforcement learning.

1 INTRODUCTION

Model distillation (Hsieh et al., 2023) transfers capabilities through supervised fine-tuning (SFT) of a student on teacher responses. For example, responses from GPT-4o and GPT-4o-mini (OpenAI, 2024a) were used for fine-tuning OLMo 2 (Walsh et al., 2025) and SARDI-Dream-7B (Jünger et al., 2026), respectively. However, providers now prohibit using such outputs to train competing models (Anthropic, 2025a; OpenAI, 2026b) and have reported attempts to collect them for that purpose (Anthropic, 2026a; OpenAI, 2026a). Given that released artifacts alone do not reveal whether a student was trained on such data, recent work on distillation attribution (Wadhwa et al., 2025; Rawat et al., 2026) seeks to identify which *teacher candidate* generated the responses used to train the student. For attribution to serve as credible provenance evidence, an audit must rely only on externally available information and clarify the evidentiary scope of positive and negative verdicts.

Existing methods either compare token likelihoods under the student and a pre-distillation checkpoint (Rawat et al., 2026) or train a classifier on Part-of-Speech (PoS) n-grams (Wadhwa et al., 2025). In this paper, we leverage this *syntactic signal* by aggregating patterns that recur across each candidate's responses and filtering those that no candidate uses distinctively. Even after filtering, a candidate may appear close to a student simply because it sits close to other candidates' text in general, or because its distances vary widely enough that such closeness is unremarkable. To separate unusual proximity from these tendencies, we additionally use a candidate-pool calibration that evaluates each student–candidate distance relative to distances between that candidate and other candidates. We call the resulting method SCOUT (**S**yntactic **C**alibration for **O**utput-only **U**nreferenced **T**argets), which only uses current text outputs from the student and teacher candidates, requiring no classifier, model weights, token likelihoods, or historical checkpoints.

The need for such an output-only method becomes particularly clear in the *retrospective* setting we introduce, where distillation attribution is performed after the audited model has been released and may have undergone further training phases such as SFT, preference optimization, or reinforcement learning, possibly by a third party. In this setting, subsequent training may alter the observable teacher-associated signature. Moreover, a pre-distillation checkpoint may be unavailable, and mis-

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.

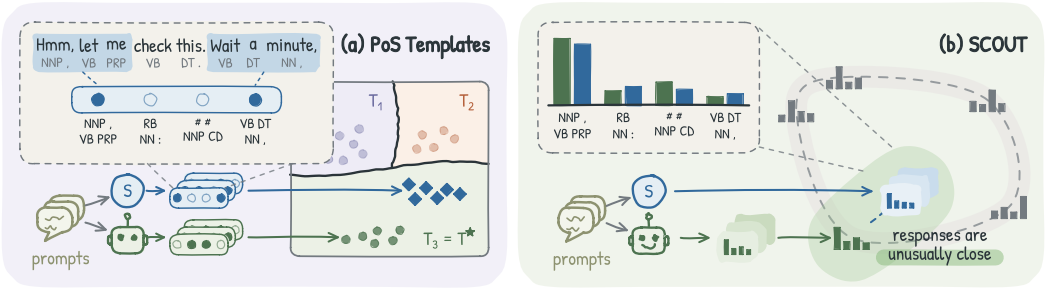


Figure 1: **How PoS Templates and SCOUT use syntactic evidence for distillation attribution. (a)** PoS Templates learns a classifier from syntactic patterns in individual candidate responses and classifies each student's responses. **(b)** SCOUT aggregates recurring syntactic patterns across each teacher candidate's responses into a *profile* and compares each student response with these profiles.

takenly using a post-distillation checkpoint can mask the distillation effect. An external auditor may also observe only current text outputs. Consequently, we ask whether student and candidate outputs retain sufficient evidence for distillation attribution and abstention. Our contributions are as follows:

Method: We propose SCOUT, an output-only distillation attribution method that aggregates patterns across prompts into candidate profiles, filters low-contrast patterns, and calibrates student–candidate distances against the pool to identify an unusually close candidate. Figure 1 contrasts SCOUT with PoS Templates (Wadhwa et al., 2025).

Retrospective Evaluation: We evaluate distillation attribution on public descendants of models distilled from DeepSeek-R1 (Guo et al., 2025) and subsequently trained by different groups through additional SFT, preference optimization, or reinforcement learning. SCOUT selects the documented teacher for all evaluated descendants within the candidate pool using current outputs alone.

Findings: Using SCOUT as an analytical instrument, we observe that syntactic teacher signatures emerge during SFT and remain readable through preference optimization and reinforcement learning across public training trajectories and released descendants spanning dense and mixture-of-experts architectures. Moreover, the signature emerges and persists in a diffusion language model lineage (Ye et al., 2025), suggesting that the phenomenon is not confined to autoregressive generation.

2 Related Work

Distillation Attribution and Model Provenance PoS Templates (Wadhwa et al., 2025) trains a classifier on syntactic patterns and uses it to assign each student response to one of the teacher candidates, but has no native rejection rule. DistillDetect (Rawat et al., 2026) ranks candidates by likelihood shifts from an earlier student checkpoint. DLI (Zeng et al., 2026) verifies a hypothesized teacher, but its text-only setting still requires shadow distillation and auditor training. Separate model-provenance work tests parameter ancestry among fine-tuned or quantized descendants (Foley et al., 2023; Yax et al., 2025; Nikolic et al., 2025; Shao et al., 2026; Hu et al., 2026), rather than which teacher generated the responses used for SFT. SCOUT addresses teacher attribution after subsequent training using only current student and candidate texts. It adapts reference-population normalization from speaker verification, stylometry, and membership inference (Rosenberg et al., 1992; Burrows, 2002; Watson et al., 2022) and abstains when the evidence distinguishes no candidate.

Proactive Model Protection Proactive defenses monitor query streams (Juuti et al., 2019), modify models through active fingerprints (Xu et al., 2024), watermark generations (Kirchenbauer et al., 2023; Sander et al., 2024; 2026), or alter sampling, interaction behavior, and reasoning traces to impede distillation (Savani et al., 2025; Yang et al., 2026; Ma et al., 2026). Weight-release defenses transform checkpoints to resist unauthorized merging (Junhao et al., 2025; Wang et al., 2025; Jang et al., 2026). These mechanisms can require monitoring infrastructure, additional training, controlled decoding, or intervention in served responses. They may add serving overhead or introduce utility trade-offs. Moreover, they must be deployed while the owner controls the model or API. They cannot protect outputs collected without these safeguards or address suspicions that arise only

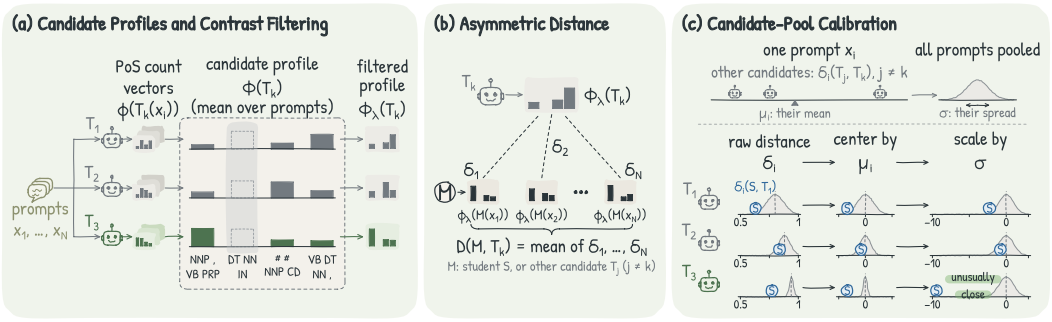


Figure 2: **Overview of SCOUT. (a)** PoS n-grams are aggregated into candidate profiles and low-contrast patterns are filtered. **(b)** Each response is compared with the filtered profiles. **(c)** Inter-candidate distances provide prompt-specific centers μ_i and candidate-level scales σ. Filtering isolates candidate-discriminative syntax, while calibration makes scores comparable across candidates. Therefore, low scores indicate proximity not explained by generic similarity to the pool.

after a third-party model is released. In such cases, post-hoc auditing becomes necessary. SCOUT addresses this complementary need without modifying the audited or candidate models.

3 Problem Formulation

Training Trajectory For any model M and prompt x, let $M(x)$ be the response produced by M for x. Let $T^\star$ denote the *teacher model* whose outputs constitute the data source for model distillation, and let $S^{(0)}$ denote the student checkpoint immediately before the distillation step. For a set of training prompts $\mathcal{Q}$, the supervision dataset is $\mathcal{D}^\star = \{(q, T^\star(q)) \mid q \in \mathcal{Q}\}$. SFT on $\mathcal{D}^\star$ produces $S^{(1)}$, after which the student may undergo further training:

$$S^{(0)} \xrightarrow{\text{SFT on } \mathcal{D}^\star} S^{(1)} \xrightarrow{\mathcal{P}_1} \cdots \xrightarrow{\mathcal{P}_{\ell-1}} S^{(\ell)} \xrightarrow{\mathcal{P}_\ell} S^{(\ell+1)} \xrightarrow{\mathcal{P}_{\ell+1}} \cdots \xrightarrow{\mathcal{P}_{L-1}} S^{(L)}. \quad (1)$$

Here, each $\mathcal{P}_\ell$ denotes the ℓ-th downstream training stage, such as additional SFT, preference optimization, or reinforcement learning. We write $S := S^{(L)}$ for the released model available to the auditor. If no training follows distillation, intermediate terms are omitted, i.e., $L = 1$ and $S = S^{(1)}$. Retrospective distillation attribution targets $T^\star$, the teacher model for the step from $S^{(0)}$ to $S^{(1)}$.

Auditor Observations Let $\mathcal{T} = \{T_k\}_{k=1}^K$ be the teacher candidate pool, which may or may not contain $T^\star$. Given audit prompts $\mathcal{X} = \{x_i\}_{i=1}^N$, the auditor queries S and every $T_k \in \mathcal{T}$ on each prompt x_i. The auditor has access only to the text responses and cannot assume access to model weights, token likelihoods, the supervision data $\mathcal{D}^\star$, or any earlier checkpoint $S^{(\ell)}$ with $0 \le \ell < L$.

Attribution and Abstention The auditor compares responses from S and each $T_k \in \mathcal{T}$ on shared audit prompts $\mathcal{X}$, returning a verdict $\widehat{T} \in \mathcal{T} \cup \{\perp\}$, where $\perp$ denotes abstention. Correctness requires $\widehat{T} = T^\star$ when $T^\star \in \mathcal{T}$, and $\widehat{T} = \perp$ when $T^\star \notin \mathcal{T}$. Exact recovery is the objective, but resolution depends on the pool: successive releases or models connected by distillation may produce outputs too similar for reliable exact attribution. Moreover, a verdict of $\perp$ means only that current evidence supports no candidate, not that none supplied responses at an earlier training stage.

4 Proposed Method

In this section, we introduce SCOUT, an output-only distillation attribution method that combines (a) PoS n-gram candidate profiles, (b) candidate-contrast filtering, and (c) candidate-pool calibration for attribution with abstention. Figure 2 summarizes these components.

4.1 Response Representation and Candidate Profiles

To capture linguistic form rather than content or task correctness, we encode a response $M(x)$ as the count vector $\phi(M(x))$ of contiguous PoS n-grams for $n \in \{3, 4, 5\}$. Replacing words with syntactic tags reduces lexical sensitivity while retaining local structure (Shaib et al., 2024; Wadhwa

et al., 2025). However, a single response realizes the patterns present in one generation only. Consequently, rather than fitting a classifier to individual response vectors, we represent each candidate T_k by its centroid across N prompts, termed its *candidate profile*:

$$\Phi(T_k) = \frac{1}{N}\sum_{i=1}^{N} \phi\big(T_k(x_i)\big). \tag{2}$$

$\Phi(T_k)$ captures recurring patterns across candidate responses, while the audited model remains at the prompt level. Appendix A provides examples in model responses.

4.2 Candidate-Contrast Filtering

Patterns used at similar rates by every candidate can dominate frequency-based similarity while contributing little to candidate discrimination. We identify such patterns from the candidate profiles themselves. First, we normalize each candidate profile by its total count:

$$\widetilde{\Phi}(T_k) = \frac{\Phi(T_k)}{\|\Phi(T_k)\|_1}. \tag{3}$$

Then, over features observed in at least one candidate, we compute the coordinatewise mean $\bar{\Phi}$, standard deviation s_Φ, and contrast vector c:

$$\bar{\Phi} = \frac{1}{K}\sum_{k=1}^{K} \widetilde{\Phi}(T_k), \qquad s_\Phi = \left[\frac{1}{K}\sum_{k=1}^{K}\left(\widetilde{\Phi}(T_k) - \bar{\Phi}\right)^2\right]^{\frac{1}{2}}, \qquad c = \frac{s_\Phi}{\bar{\Phi}}. \tag{4}$$

The square, square root, and division are coordinatewise. Small entries of c indicate patterns shared at similar rates across candidates, whereas large entries indicate candidate-specific variation.

Finally, let $\lambda \in [0, 1]$ denote the filtering fraction and γ_λ the λ-quantile of c over features observed in the current candidate pool. The mask $w_\lambda = \mathbf{1}[c \geq \gamma_\lambda]$ retains features at or above this cutoff. Features absent from every candidate are excluded from the quantile and retain unit weight. Thus, $\lambda = 0$ recovers the unfiltered representation, while increasing λ removes more low-contrast features. Using this mask, we define $\phi_\lambda(y) = w_\lambda \odot \phi(y)$ and $\Phi_\lambda(T_k) = w_\lambda \odot \Phi(T_k)$, where $\odot$ denotes the Hadamard product. We fix $\lambda = 0.15$ for all experiments and report sensitivity to λ in Appendix B.

4.3 Candidate-Pool Calibration and Decision Rule

Candidate-Pool Calibration For $M \in \{S\} \cup \mathcal{T}$, we use the filtered representations to define the prompt-level profile distance and its uncalibrated aggregate as

$$\delta_i(M, T_k) = 1 - \cos(\phi_\lambda(M(x_i)), \Phi_\lambda(T_k)), \qquad D(M, T_k) = \frac{1}{N}\sum_{i=1}^{N} \delta_i(M, T_k). \tag{5}$$

Lower values indicate greater syntactic similarity, but raw distances are not comparable across candidates as their baseline proximity and dispersion can differ. SCOUT calibrates them using null locations and scales estimated from inter-candidate distances. Treating every other candidate T_j as a pseudo-suspect, SCOUT estimates the prompt-specific null location and candidate-level null scale

$$\mu_i(T_k) = \frac{1}{K-1}\sum_{j \neq k} \delta_i(T_j, T_k), \qquad \sigma(T_k) = \mathrm{std}\{\delta_i(T_j, T_k) : j \neq k,\ 1 \leq i \leq N\}. \tag{6}$$

Here, std is the population standard deviation over the $(K-1)N$ distances. Prompt-specific means capture response structure, while one scale per candidate pools all prompts because each prompt provides only $K-1$ pseudo-suspects. The centered and standardized scores are

$$C(S, T_k) = \frac{1}{N}\sum_{i=1}^{N} [\delta_i(S, T_k) - \mu_i(T_k)], \qquad Z(S, T_k) = \frac{C(S, T_k)}{\sigma(T_k)}. \tag{7}$$

Centering and scaling normalize response similarity relative to the candidate pool, so lower $Z(S, T_k)$ means that the student is closer to T_k than the pool null predicts. This normalization allows SCOUT to use only current student and candidate responses, without classifier training, model weights, token likelihoods, or historical checkpoints. Appendix C formalizes the correction, derives finite-prompt bounds, and characterizes limits of candidate-only calibration and output-only history recovery.

Attribution and Abstention For closed-set attribution, the auditor selects

$$\widehat{T} = \arg\min_{T_k \in \mathcal{T}} Z(S, T_k). \tag{8}$$

For open-set attribution, we define the closest-alternative gap:

$$G(S) = \min_{T' \in \mathcal{T} \setminus \{\widehat{T}\}} Z(S, T') - Z(S, \widehat{T}). \tag{9}$$

Larger $G(S)$ indicates clearer separation. The auditor accepts $\widehat{T}$ when $G(S) \geq \tau$ and abstains otherwise, with τ fitted without the audited fold, following Section 5.3.

4.4 Trajectory Margins

For the trajectory analysis in Section 6, the designated teacher T and intermediate checkpoints $S^{(\ell)}$ are known. We summarize each checkpoint using the pool-average teacher gap

$$G_T\big(S^{(\ell)}\big) = \frac{1}{K-1} \sum_{T' \in \mathcal{T} \setminus \{T\}} \Big[Z\big(S^{(\ell)}, T'\big) - Z\big(S^{(\ell)}, T\big) \Big]. \tag{10}$$

Larger values indicate stronger separation toward T relative to the rest of the pool. Averaging over alternatives prevents shifts in the closest rival from appearing as movement toward or away from T.

For each checkpoint, we report the change in this margin toward T:

$$\Delta G_T\big(S^{(\ell)}\big) = G_T\big(S^{(\ell)}\big) - G_T\big(S^{(0)}\big). \tag{11}$$

Section 6 uses a common candidate pool, with candidate profiles and empirical nulls fixed along each trajectory. Positive values indicate stronger teacher-relative separation than at $S^{(0)}$. Both G_T and ΔG_T are teacher-aware descriptive quantities and do not determine attribution verdicts.

5 Experiments

In this section, we evaluate SCOUT in two regimes. In Section 5.2, we evaluate it retrospectively on released descendants of DeepSeek-R1 (Guo et al., 2025) distillations built on Qwen2.5-Math bases (Yang et al., 2024a) after further post-training, which may have altered the teacher's syntactic signature. Section 5.3 evaluates it in the controlled setting, where students are audited immediately after SFT and the known teacher can be included in or removed from the candidate pool.

5.1 Compared Methods and Probe Sets

We compare SCOUT with DistillDetect (Rawat et al., 2026) and PoS Templates (Wadhwa et al., 2025). Because PoS Templates is a closed-set classifier without native abstention, we retrain it on the three remaining candidates in teacher-removed controlled trials and use its top-two class-probability gap as the detection score. Following Zheng et al. (2023), we also evaluate two training-free LLM judges, Mistral Medium 3.1 (Mistral AI, 2025) and Solar Pro 4 (Upstage, 2026). Score-based attribution follows the DistillDetect evaluator's mean-score rule (Rawat et al., 2026), while controlled detection also uses its threshold and significance procedures. LLM judges use the most frequent prompt-level label, with ties yielding no verdict. Appendix D reports component comparisons and analyzes distillation attribution under a sibling-enriched candidate pool.

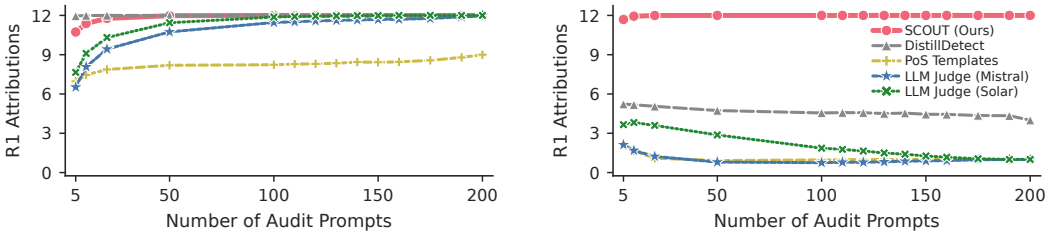
We use four probe sets throughout the evaluation. OpenMathInstruct (OMI) (Toshniwal et al., 2024) and s1 (Muennighoff et al., 2025b) serve as math probes, while OASST1 (Köpf et al., 2023) and Databricks Dolly 15K (Dolly) (Conover et al., 2023) serve as conversation probes. Appendix E documents the public checkpoints, their designated teachers, and training sequences used in the retrospective and trajectory analyses, while Appendix F provides additional experimental details, including the prompts provided to the LLM judges.

5.2 Distillation Attribution After Subsequent Training

Evaluation Panel We study DeepSeek-R1-Distill-Qwen-1.5B and 7B distilled from DeepSeek-R1, whose pre-distillation bases are Qwen2.5-Math-1.5B and 7B, respectively. We audit these two bases, two distilled parents, and their twelve public descendants. As detailed in Appendix F, ten descendants underwent reward-based or self-play post-training without additional R1 responses, whereas the remaining two were re-exposed to R1 responses during further training. Since R1 is the teacher for every distilled parent and descendant, this panel tests whether its signature remains detectable after subsequent training and absent from the pre-distillation bases.

Table 1: **Distillation attribution after subsequent training.** *Desc.* and *Par.* count correct R1 assignments among twelve descendants and two distilled parents, while *Base* counts pre-distillation bases correctly identified as non-R1. Using the same twelve descendants and prompt bank, *Dist.* compares independently sorted candidate scores at equal ranks, whereas *Same* compares candidates on each shared prompt. Both report probe-level R1 selection accuracy averaged across descendants, distinguishing distributional separation from same-prompt candidate selection. For DistillDetect, R denotes the reference model. *Dist.* is unavailable for LLM judges because they return labels only. Higher is better for all metrics. Bold marks the best output-only value in each column.

Math probes	OMI					s1				
	Model Level			Probe Level (%)		Model Level			Probe Level (%)	
Method	Desc.	Par.	Base	Dist.	Same	Desc.	Par.	Base	Dist.	Same
Reference-based (A pre-distillation reference checkpoint is required.)										
DistillDetect, $R = S^{(0)}$	12/12	2/2	2/2	98.0	94.1	12/12	2/2	2/2	97.9	96.2
DistillDetect, $R = S^{(1)}$	3/12	—	—	—	—	3/12	—	—	—	—
Output-only (Only current model outputs are required.)										
PoS Templates	9/12	1/2	**2/2**	66.6	42.1	9/12	**2/2**	**2/2**	79.4	46.9
LLM judge (Mistral)	**12/12**	**2/2**	**2/2**	—	45.5	11/12	**2/2**	**2/2**	—	42.0
LLM judge (Solar)	**12/12**	**2/2**	**2/2**	—	53.3	**12/12**	**2/2**	1/2	—	54.2
SCOUT (Ours)	**12/12**	**2/2**	**2/2**	**87.9**	**70.8**	**12/12**	**2/2**	**2/2**	**99.0**	**80.1**
Conversation probes	OASST1					Dolly				
	Model Level			Probe Level (%)		Model Level			Probe Level (%)	
Method	Desc.	Par.	Base	Dist.	Same	Desc.	Par.	Base	Dist.	Same
Reference-based (A pre-distillation reference checkpoint is required.)										
DistillDetect, $R = S^{(0)}$	0/12	0/2	2/2	6.3	36.0	4/12	1/2	2/2	28.3	43.5
DistillDetect, $R = S^{(1)}$	2/12	—	—	—	—	2/12	—	—	—	—
Output-only (Only current model outputs are required.)										
PoS Templates	1/12	0/2	**2/2**	7.8	25.3	1/12	0/2	**2/2**	7.7	22.1
LLM judge (Mistral)	0/12	0/2	**2/2**	—	19.4	1/12	0/2	**2/2**	—	27.7
LLM judge (Solar)	1/12	0/2	1/2	—	29.8	1/12	0/2	0/2	—	34.9
SCOUT (Ours)	**12/12**	**2/2**	**2/2**	**92.9**	**75.1**	**12/12**	**2/2**	**2/2**	**95.5**	**84.5**



(a) OMI probe set

(b) Dolly probe set

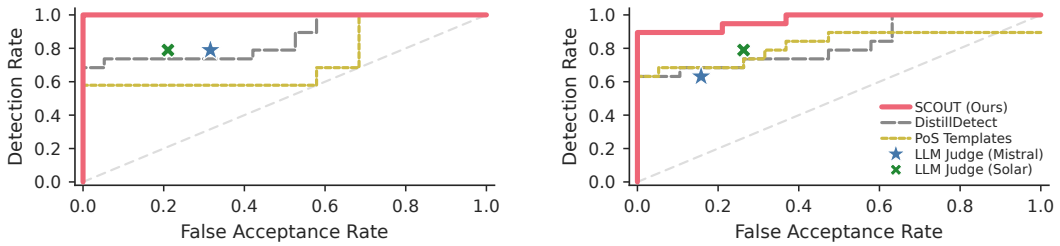
Figure 3: **Prompt efficiency after subsequent training.** For each audit budget $n < 200$, we randomly select n distinct prompts 2,000 times and report the mean number of the 12 descendants correctly attributed to R1. At $n = 200$, we evaluate all prompts once. Higher is better.

Attribution Performance We report model-level accuracy together with two probe-level accuracies that measure distributional separation and same-prompt candidate selection, respectively. Table 1 shows that SCOUT correctly assigns all descendants and parents to R1 and identifies both pre-distillation bases as non-R1 on all four probes. Under both probe-level views, it attains the highest output-only accuracy on every probe, indicating that its R1 signal appears both in the score distribution and in same-prompt comparisons. With $S^{(0)}$, DistillDetect matches SCOUT on all math model-level counts. Using the already-distilled parent $S^{(1)}$ instead yields only $3/12$ descendants on OMI and $3/12$ on s1 because this comparison reflects subsequent post-training rather than the R1 distillation step. Model-level attribution uses distinct ten-candidate pools for math and conversation.

Prompt Efficiency Figure 3 evaluates prompt efficiency on OMI and Dolly. With five prompts, SCOUT attributes $10.7/12$ descendants on OMI and $11.7/12$ on Dolly, reaching $12/12$ with 100 and 50 prompts, respectively. On OMI, it approaches DistillDetect with $S^{(0)}$ and is more efficient than the other output-only methods. On Dolly, SCOUT remains near-perfect as the budget grows, whereas DistillDetect declines despite access to $S^{(0)}$ and the other output-only methods fall to $1/12$.

Table 2: **Detection and threshold transfer.** Both 1.5–4B cohorts use disjoint SFT and audit examples. Math uses OMI and s1, and conversation uses OASST1 and Dolly. Det. counts students correctly attributed on both probes with the teacher present using student-backbone holdout. $\mathrm{Acc}(S)$ and $\mathrm{Acc}(T)$ count correct accept-or-abstain decisions under student-backbone and teacher holdout, respectively, across 4 cells per student and 76 per cohort. $\mathrm{FP}(S)$ and $\mathrm{FP}(T)$ count accepted students after teacher removal under the same holdouts. Higher is better for Det. and Acc, and lower for FP.

	Math (19 Students)					Conversation (19 Students)				
Method	Det.	$\mathrm{Acc}(S)$	$\mathrm{Acc}(T)$	$\mathrm{FP}(S)$	$\mathrm{FP}(T)$	Det.	$\mathrm{Acc}(S)$	$\mathrm{Acc}(T)$	$\mathrm{FP}(S)$	$\mathrm{FP}(T)$
Reference-based (A pre-distillation reference checkpoint is required.)										
DistillDetect	11/19	57/76	43/76	3/19	10/19	13/19	54/76	38/76	5/19	12/19
Output-only (Only current model outputs are required.)										
PoS Templates	14/19	48/76	32/76	11/19	11/19	11/19	64/76	53/76	**0/19**	**0/19**
LLM judge (Mistral)	15/19	56/76	56/76	6/19	6/19	12/19	56/76	56/76	3/19	3/19
LLM judge (Solar)	15/19	54/76	54/76	4/19	4/19	**15/19**	58/76	58/76	5/19	5/19
SCOUT (Ours)	**18/19**	**74/76**	58/76	**0/19**	**0/19**	**15/19**	**68/76**	**69/76**	**0/19**	**0/19**
− Scaling (C)	16/19	68/76	57/76	**0/19**	**0/19**	14/19	64/76	64/76	2/19	2/19
− Centering (D)	17/19	73/76	**68/76**	**0/19**	**0/19**	12/19	61/76	65/76	2/19	2/19



(a) Math cohort (b) Conversation cohort

Figure 4: **Detection performance across false-acceptance rates.** Axes use the student-level definitions in Table 2. LLM judges appear as single operating points, and the diagonal denotes chance. SCOUT matches the best math performance and is best in conversation at low false-acceptance rates.

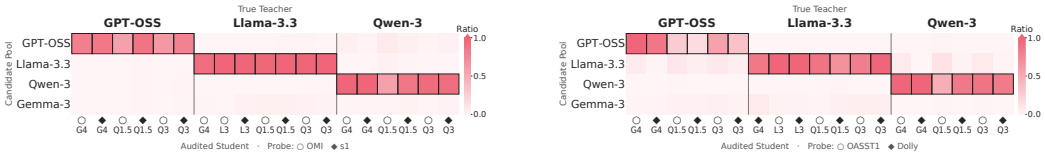
5.3 Distillation Attribution Before Subsequent Training

Controlled Cohorts and Protocol Each cohort contains nineteen 1.5–4B students based on Gemma-3-4B-PT (Gemma Team et al., 2025), Llama-3.2-3B-Instruct (Grattafiori et al., 2024), and Qwen-2.5 at 1.5B and 3B (Yang et al., 2024b). Across matched configurations, the teacher varies among GPT-OSS-120B (OpenAI, 2025a), Llama-3.3-70B-Instruct, and Qwen-3-8B (Yang et al., 2025a). Gemma-3-27B-it completes the four-candidate pool as an unexposed distractor. Identification uses the probe set not used for SFT, whereas detection and abstention use both probe sets.

For score-based methods, we fit the acceptance threshold on the non-held-out cells of each fold and apply it to the held-out group. SCOUT applies this threshold to the winning margin $G(S)$ defined in Section 4.3. Student-backbone holdout excludes all cells sharing one backbone, while teacher holdout excludes all cells associated with one teacher and tests transfer to a teacher unseen during threshold fitting. We denote these schemes by S and T, respectively, following Rawat et al. (2026).

Attribution, Abstention, and Calibration Table 2 shows that, under the fold-fitted thresholds, SCOUT detects $18/19$ math students and $15/19$ conversation students while accepting none after teacher removal. Moving from D to C reduces detection by one student in math but raises it by two in conversation. Adding scaling improves detection in both and eliminates false positives in conversation. Thus, gains are not monotonic across stages, but full location-and-scale calibration gives the strongest attribution-abstention balance across domains.

Figure 4 shows that this separation is not specific to the fitted thresholds. SCOUT attains student-level AUCs of 1.000 and 0.970, respectively. At zero false positives, it detects $19/19$ and $17/19$ students, compared with $13/19$ and $12/19$ for DistillDetect, which requires a pre-distillation reference checkpoint. Appendix F describes the controlled cohorts, while Appendix D reports the full calibration ablation, including a 7–8B cohort testing whether the results extend to larger students.



(a) Math cohort (b) Conversation cohort

Figure 5: **Controlled students reproduce teacher-associated syntactic patterns.** Rows are candidate models, and columns are 1.5–4B students grouped by their true teacher. Each cell reports the share of student PoS n-gram tokens matching the candidate's ten most characteristic patterns, normalized by the share in responses from that candidate. Column labels use G4 for Gemma-3-4B-PT, L3 for Llama-3.2-3B-Instruct, Q1.5 for Qwen-2.5-1.5B, and Q3 for Qwen-2.5-3B.

Inheritance of Syntactic Signatures Figure 5 visualizes the syntactic evidence captured by profile distance. For each candidate, we identify its ten most characteristic PoS n-grams relative to the pool and measure their frequency in held-out student responses, normalized by candidate responses. The block-diagonal structure across backbones and domains shows consistent inheritance of teacher-associated patterns. Appendix A shows response examples, while Appendix B provides full-size heatmaps and further inheritance analyses.

Effect of Candidate-Specific Scaling Figure 6 compares raw, centered, and scaled decision gaps for ten students on one held-out Dolly prompt. Raw gaps correctly attribute $8/10$ students, while centered gaps attribute $7/10$. The three centered errors remain within 0.007 of zero because centering removes baseline proximity but not differences in candidate scale. Dividing by the candidate-specific spread σ moves all ten scaled gaps above zero. This single-prompt example illustrates the correction. The full procedure aggregates 200 prompts per probe and requires agreement across probes.

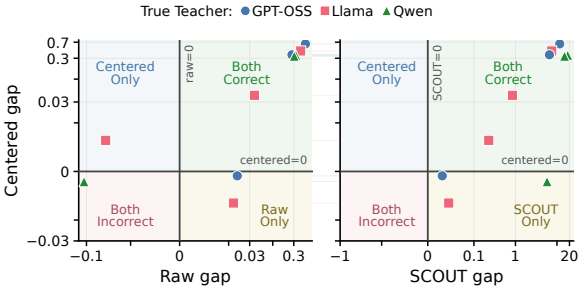


Figure 6: **Scaling corrects errors after centering.** For ten students on one held-out Dolly prompt, panels plot raw against centered gaps and centered against scaled gaps. Positive gaps indicate correct attribution.

Takeaway 1: SCOUT Enables Output-Only Attribution Across Training Stages.

In controlled cohorts audited immediately after SFT, SCOUT achieves high detection and accepts no candidate after the teacher is removed from the pool. After subsequent training, it attributes every public descendant to the documented teacher and identifies both pre-distillation bases as non-R1 using only current outputs, without a pre-distillation checkpoint.

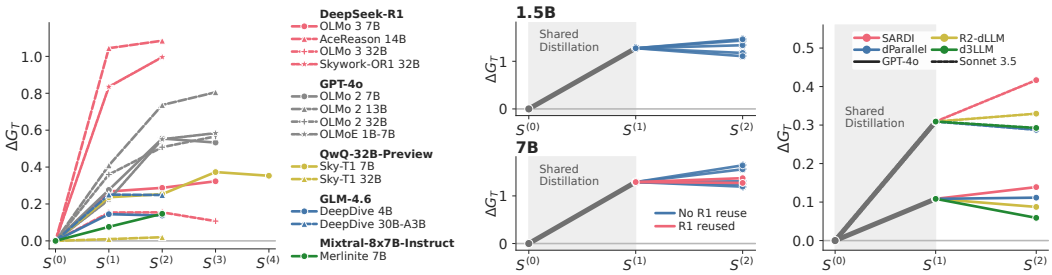
6 Reading the Signature Along a Training Trajectory

The preceding section evaluates SCOUT before and after subsequent training. In this section, we use the trajectory margin change ΔG_T defined in Eq. 11 to examine when teacher-associated evidence emerges and whether it persists through subsequent training.

6.1 Teacher Signatures Emerge and Persist Across Training Trajectories

First, we examine thirteen public trajectories spanning five teachers, dense and mixture-of-experts architectures, and roughly 1B active to 32B dense parameters. Figure 7a shows that, on OMI, ΔG_T becomes positive immediately after distillation and remains positive at every later checkpoint in all thirteen trajectories. The same pattern holds on s1. Appendix E reports the s1 results and documents each trajectory, its designated teacher, and its training data.

Furthermore, we examine the two R1-distilled parents and twelve public descendants from Section 5.2. For each lineage, the Qwen2.5-Math base is $S^{(0)}$, the R1-distilled parent is $S^{(1)}$, and each



(a) Thirteen public ladders (b) Qwen2.5-Math lineages (c) Dream-7B lineages

Figure 7: **Source-margin shifts from the pre-distillation state on OMI.** All panels use the same 13-candidate pool. Positive values indicate a larger margin toward the indicated teacher than at $S^{(0)}$, and every post-distillation value is positive. Shading marks distillation in **(b)** and **(c)**.

descendant is a subsequent training outcome $S^{(2)}$. Figure 7b shows $\Delta G_T > 0$ on OMI for both parents and all twelve descendants. The same holds on s1, so every descendant retains an R1 signature relative to $S^{(0)}$. Compared with its corresponding $S^{(1)}$ parent, the margin is higher for $7/12$ descendants on OMI and $3/12$ on s1. Thus, the signature persists across the panel, while further strengthening varies among descendants.

6.2 The Signature Extends Beyond Autoregressive Models

Because SCOUT compares contiguous PoS n-grams, the signal could depend on autoregressive token generation. Therefore, we apply the same readout to Dream-7B (Ye et al., 2025), a diffusion language model, evaluating its pre-distillation base, instruction-tuned successor, and four public post-training descendants. Since its instruction data include responses from both GPT-4o (OpenAI, 2024a) and Sonnet 3.5 (Anthropic, 2024), Figure 7c reports separate margins toward the two teacher models. Both margins increase after instruction tuning and remain positive across all four branches on both probes, showing that signatures associated with both documented teachers can emerge and persist in a diffusion language model.

Takeaway 2: Teacher Signatures Remain Readable Through Diverse Post-Training.

SCOUT traces teacher-associated syntactic signatures that emerge during distillation and remain readable after preference optimization and reinforcement learning across dense and mixture-of-experts autoregressive models. In the diffusion model, margins toward both documented contributors remain positive under mixed-teacher supervision.

7 Conclusion

In this paper, we introduce SCOUT, an output-only method for distillation attribution. It aggregates recurring syntactic patterns into candidate profiles, filters low-contrast patterns, and calibrates student–candidate distances against inter-candidate distances. Across controlled students and public descendants, SCOUT supports open-set attribution in controlled cohorts and closed-set retrospective attribution after subsequent training. Across training trajectories, teacher-associated signatures emerge during distillation and persist through preference optimization and reinforcement learning. The same signal extends to a diffusion language model under mixed-teacher supervision.

Limitations SCOUT depends on the supplied candidate pool, and principled pool construction for third-party auditors remains open. Where applicable, we follow the pool construction used by DistillDetect (Rawat et al., 2026), but this protocol does not determine which models to include in new settings. Pool composition also bounds attribution resolution. A sibling-enriched pool sharply reduces exact attribution from current outputs, as detailed in Appendix D.

Future Work Promising directions for future work include testing whether the observed persistence extends to watermarks and fingerprints under matched post-training and response-rewriting interventions, and evaluating adversaries optimized to evade attribution.

AI USE STATEMENT

Generative AI tools were used during manuscript preparation to provide feedback, assist literature search, and improve clarity. LLMs were also used experimentally to generate training and audit responses and serve as judge baselines, as described in Section 5 and Appendix F. These experimental roles followed the stated procedures, while interpretation and scientific conclusions remained with the authors. All AI-assisted manuscript content was reviewed and verified by the authors, who take full responsibility for the final content and claims.

ETHICS STATEMENT

Output-based attribution can inform provenance audits, but should not be treated as conclusive evidence of a model's training history. For example, close source relatives can reduce exact-teacher attribution. Appendix C shows that subsequent SFT can make an earlier source unrecoverable by the evaluated methods under the tested training budgets and probe sets. This result has dual-use implications because it may inform attempts to evade attribution. It also shows that non-detection does not establish non-use. Therefore, attribution findings should be assessed alongside independent provenance evidence.

REPRODUCIBILITY STATEMENT

The response representation is specified in Section 4, with candidate-pool calibration in Eqs. 6–7. Appendix B reports preprocessing, filtering, and robustness analyses, Appendix C gives the finite-prompt analysis, and Appendix F documents generation, training, and evaluation procedures.

REFERENCES

Anthropic. Claude 3.5 sonnet model card addendum, 2024. URL https://www-cdn.anthropic.com/fed9cc193a14b84131812372d8d5857f8f304c52/Model_Card_Claude_3_Addendum.pdf.

Anthropic. Commercial terms of service. Anthropic Legal, June 2025a. URL https://www.anthropic.com/legal/commercial-terms. Effective 17 June 2025. Accessed 14 August 2026.

Anthropic. System card: Claude opus 4 & claude sonnet 4, 2025b. URL https://www-cdn.anthropic.com/4263b940cabb546aa0e3283f35b686f4f3b2ff47.pdf.

Anthropic. Claude opus 4.5, 2025c. URL https://www.anthropic.com/news/claude-opus-4-5.

Anthropic. Detecting and preventing distillation attacks. Anthropic News, February 2026a. URL https://www.anthropic.com/news/detecting-and-preventing-distillation-attacks. Published 23 February 2026. Accessed 14 August 2026.

Anthropic. Claude opus 4.6, 2026b. URL https://www.anthropic.com/news/claude-opus-4-6.

Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pp. 214–217, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/P04-3031/.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

John Burrows. 'delta': a measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing*, 17(3):267–287, 09 2002.

Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron: Advancing math and code reasoning through reinforcement learning, 2025. URL https://arxiv.org/abs/2505.16400.

Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. dparallel: Learnable parallel decoding for dLLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hVOcstAURb.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Zhenbang Du, Kejing Xia, Xinrui Zhong, Yonggan Fu, Nicolai Oswald, Binfei Ji, Brucek Khailany, Pavlo Molchanov, and Yingyan Lin. r^2-dllm: Accelerating diffusion large language models via spatio-temporal redundancy reduction, 2026. URL https://arxiv.org/abs/2604.18995.

Falcon-LLM Team. The falcon 3 family of open models, December 2024. URL https://huggingface.co/blog/falcon3.

Myles Foley, Ambrish Rawat, Taesung Lee, Yufang Hou, Gabriele Picco, and Giulio Zizzo. Matching pairs: Attributing fine-tuned models to their pre-trained large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7423–7442, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.410. URL https://aclanthology.org/2023.acl-long.410/.

Gemma Team et al. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

GLM-4.5 Team et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025. URL https://arxiv.org/abs/2508.06471.

Aaron Grattafiori et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Daya Guo et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner 1 technical report, 2025. URL https://arxiv.org/abs/2505.22312.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 01621459, 1537274X. URL http://www.jstor.org/stable/2282952.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

Yuepeng Hu, Zhengyuan Jiang, Mengyuan Li, Osama Ahmed, Zhicong Huang, Cheng Hong, and Neil Zhenqiang Gong. Fingerprinting LLMs via prompt injection. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11795–11810, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.541. URL https://aclanthology.org/2026.acl-long.541/.

Minwoo Jang, Hoyoung Kim, Jabin Koo, and Jungseul Ok. Making models unmergeable via scaling-sensitive loss landscape. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=o9NThz6auq.

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

Albert Q. Jiang et al. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Paul Jünger, Justin Lovelace, Linxi Zhao, Dongyoung Go, and Kilian Q Weinberger. Self-augmenting retrieval for diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=uiE6BIx2vC.

Wei Junhao, Yu Zhe, and Jun Sakuma. Disrupting model merging: A parameter-level defense without sacrificing accuracy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17698–17707, October 2025.

Mika Juuti, Sebastian Szyller, Samuel Marchal, and N. Asokan. Prada: Protecting against dnn model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 512–527, 2019. doi: 10.1109/EuroSP.2019.00044.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations - democratizing large language model alignment. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Gang Li, Yan Chen, Ming Lin, and Tianbao Yang. Drpo: Efficient reasoning via decoupled reward policy optimization. *arXiv preprint arXiv:2510.04474*, 2025a.

Gang Li, Ming Lin, Tomer Galanti, Zhengzhong Tu, and Tianbao Yang. Disco: Reinforcing large reasoning models with discriminative constrained optimization. *arXiv preprint arXiv:2505.12366*, 2025b.

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint arXiv:2506.24119*, 2025a. URL https://arxiv.org/abs/2506.24119.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025b. URL https://arxiv.org/abs/2505.24864.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025.

Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, et al. Exploring the limit of outcome reward for learning mathematical reasoning. *arXiv preprint arXiv:2502.06781*, 2025.

Xinhang Ma, William Yeoh, Ning Zhang, and Yevgeniy Vorobeychik. Protecting language models against unauthorized distillation through trace rewriting. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11307–11324, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.519. URL https://aclanthology.org/2026.acl-long.519/.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004/.

Mistral AI. Medium is the new large: Introducing mistral medium 3. https://mistral.ai/news/mistral-medium-3/, 2025.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. Olmoe: Open mixture-of-experts language models, 2025a. URL https://arxiv.org/abs/2409.02060.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025b. URL https://arxiv.org/abs/2501.19393.

Ivica Nikolic, Teodora Baluta, and Prateek Saxena. Model provenance testing for large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Iy4cAXotrf.

NovaSky Team. Unlocking the potential of reinforcement learning in improving reasoning models. https://novasky-ai.github.io/posts/sky-t1-7b, 2025a.

NovaSky Team. Sky-t1: Train your own o1 preview model within $450. https://novasky-ai.github.io/posts/sky-t1, 2025b.

OLMo Team. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

OpenAI. Gpt-4o system card, 2024a. URL https://arxiv.org/abs/2410.21276.

OpenAI. Openai o1 system card, 2024b. URL https://arxiv.org/abs/2412.16720.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025a. URL https://arxiv.org/abs/2508.10925.

OpenAI. Openai o3 and o4-mini system card, 2025b. URL https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf.

OpenAI. Update to the U.S. house select committee on strategic competition between the United States and the Chinese Communist Party. Letter to the U.S. House Select Committee, February 2026a. URL https://cdn.openai.com/pdf/045aa967-ee96-4a09-94ee-3098ddf6db2c/OpenAI-US-House-Select-Cmte-Update-%5B021226%5D.pdf. Dated 12 February 2026. Accessed 14 August 2026.

OpenAI. Terms of use. OpenAI Policies, 2026b. URL https://openai.com/policies/row-terms-of-use/. Accessed 14 August 2026.

Yu-Yang Qian, Junda Su, Lanxiang Hu, Peiyuan Zhang, Zhijie Deng, Peng Zhao, and Hao Zhang. d3LLM: Ultra-fast diffusion LLM using pseudo-trajectory distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=rzBAQT2Fkg.

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. URL https://qwenlm.github.io/blog/qwq-32b-preview/.

Rajat Rawat, Sizhe Chen, Akshay Anand, Michael Duan, Bob Rotsted, and Sewon Min. Reference-based distillation detection in llms, 2026. URL https://arxiv.org/abs/2607.09692.

Aaron E. Rosenberg, Joel DeLong, Chin-Hui Lee, Biing-Hwang Juang, and Frank K. Soong. The use of cohort normalized scores for speaker verification. In *2nd International Conference on Spoken Language Processing (ICSLP 1992)*, pp. 599–602, 1992. doi: 10.21437/ICSLP.1992-176.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 21079–21113. Curran Associates, Inc., 2024.

Tom Sander, Hongyan Chang, Tomáš Souček, Tuan Tran, Valeriu Lacatusu, Sylvestre-Alvise Rebuffi, Alexandre Mourachko, Surya Parimi, Christophe Ropers, Rashel Moritz, Vanessa Stark, Hady Elsahar, and Pierre Fernandez. Textseal: A localized llm watermark for provenance & distillation protection, 2026. URL https://arxiv.org/abs/2605.12456.

Yash Savani, Asher Trockman, Zhili Feng, Yixuan Even Xu, Avi Schwarzschild, Alexander Robey, Marc Anton Finzi, and J Zico Kolter. Antidistillation sampling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Vo2UHqMu8t.

Chantal Shaib, Yanai Elazar, Junyi Jessy Li, and Byron C Wallace. Detection and measurement of syntactic templates in generated text. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6416–6431, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.368. URL https://aclanthology.org/2024.emnlp-main.368/.

Shuo Shao, Yiming Li, Hongwei Yao, Yifei Chen, Yuchen Yang, and Zhan Qin. Reading between the lines: Towards reliable black-box llm fingerprinting via zeroth-order gradient estimation. In *Proceedings of the ACM Web Conference 2026*, pp. 2637–2648. ACM, April 2026. doi: 10.1145/3774904.3792196. URL http://dx.doi.org/10.1145/3774904.3792196.

Shivchander Sudalairaj, Abhishek Bhandwaldar, Aldo Pareja, Kai Xu, David D. Cox, and Akash Srivastava. Lab: Large-scale alignment for chatbots, 2024. URL https://arxiv.org/abs/2403.01081.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating AI for math with massive open-source instruction data. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=l5FDMofecw.

Upstage. Solar pro 4: The agentic model that finishes the job, 2026. URL https://www.upstage.ai/blog/en/solar-pro-4.

Somin Wadhwa, Chantal Shaib, Silvio Amir, and Byron C Wallace. Who taught you that? tracing teachers in model distillation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3307–3315, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.173. URL https://aclanthology.org/2025.findings-acl.173/.

Evan Pete Walsh et al. 2 OLMo 2 furious (COLM's version). In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=2ezugTT9kU.

Zihao Wang, Enneng Yang, Lu Yin, Shiwei Liu, and Li Shen. Model unmerging: Making your models unmergeable for secure model sharing, 2025. URL https://arxiv.org/abs/2509.01548.

Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=3eIrli0TwQ.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. Light-R1: Curriculum SFT, DPO, and RL for long CoT from scratch and beyond, 2025. URL https://github.com/Qihoo360/Light-R1.

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

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024a. URL https://arxiv.org/abs/2409.12122.

An Yang et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024b.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Guang Yang, Amir Ghasemian, Fengchen Liu, Zhong Wang, Ninareh Mehrabi, and Homa Hosseinmardi. Asking back: Interaction-layer antidistillation watermarks, 2026. URL https://arxiv.org/abs/2605.16462.

Wang Yang, Hongye Jin, Jingfeng Yang, Vipin Chaudhary, and Xiaotian Han. Thinking preference optimization, 2025b. URL https://arxiv.org/abs/2502.13173.

Nicolas Yax, Pierre-Yves Oudeyer, and Stefano Palminteri. PhyloLM: Inferring the phylogeny of large language models and predicting their performances in benchmarks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=rTQNGQxm4K.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models, 2025. URL https://arxiv.org/abs/2508.15487.

Zhirui Zeng, Jiamou Liu, Meng-Fen Chiang, Jialing He, and Shangwei Guo. Distillation lineage inspector: Black-box auditing of model distillation in LLMs, 2026. URL https://openreview.net/forum?id=mcUWhTcqTx.

Jiajie Zhang, Xin Lv, Ling Feng, Lei Hou, and Juanzi Li. Chaining the evidence: Robust reinforcement learning for deep search agents with citation-aware rubric rewards, 2026. URL https://arxiv.org/abs/2601.06021.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Zyphra. Zr1-1.5b: A small but powerful reasoning model for math and code, 2025.

Appendix Contents

A Qualitative Examples of the PoS n-gram Signal **18**

B Representation Choice and Robustness **29**

B.1 Markup and Window Controls 29

B.2 Inheritance of Teacher Signatures 30

B.3 Candidate-Contrast Sensitivity 32

B.4 Decoding and Response-Length Controls 33

C Properties and Limits of the SCOUT Readout **36**

C.1 Candidate-Relative Calibration 36

C.2 Finite-Prompt Stability 37

C.3 Limits of Candidate-Only Calibration 38

C.4 Limits of Output-Only History Recovery 40

C.5 Source Switching Under Later SFT 40

D Additional Attribution Analyses **41**

D.1 Controlled Detection Ablation 42

D.2 Candidate-Pool Stress Test with Source Relatives 42

E Additional Trajectory Analyses **43**

E.1 Thirteen Public Ladders 44

E.2 R1 Descendant Lineages 45

E.3 Masked Diffusion Lineage 45

F Experimental Details **45**

Table 3: **Candidate signatures (OMI).** The two highest-ranked non-overlapping PoS n-grams of each candidate against the mean of the other eighteen, each with one matching span.

Candidate	Signature	Example
GPT-OSS-120B (OpenAI, 2025a)	. RB NN : . NNP JJ	". Thus answer :" ". Need third"
Gemma-3-27B-it (Gemma Team et al., 2025)	$. RB $ $. PRP VBP	"$. Then $" "$. We want"
Llama-3.3-70B-Instruct (Grattafiori et al., 2024)	. # # NNP CD CD : VB DT NN	". # # Step 2" "2 : Apply the condition"
Qwen-3-8B (Yang et al., 2025a)	. NNP , VB PRP VB DT . NNP ,	". Hmm , let me" "approach this . First ,"
Claude-Sonnet-3.5 (Anthropic, 2024)	. CD) CD $ JJ $	". 2)" "8 $ * $"
Claude-Opus-4.5 (Anthropic, 2025c)	$ VB PRP VB $. VB PRP	"$ Let me compute" "$. Let me"
Claude-Opus-4.6 (Anthropic, 2026b)	# # VBG $ $ #	"# # Setting" "$ $ #"
DeepSeek-R1 (Guo et al., 2025)	NNP , NN . . NNP .	"x=0 , y=1 ." ". Hmmm ."
QwQ-32B-Preview (Qwen Team, 2024)	. VB DT NN , RB PRP VBP VBN DT	". Wait a minute ," "Maybe I have misidentified the"
o1 (OpenAI, 2024b)	NNP PRP VBP VBG VBP JJ VBG IN	"* I 'm exploring" "' m working with"
o3 (OpenAI, 2025b)	NNS VBP NN PRP VBP PRP VBP JJ VBG	"conditions * * I '" "I ' m considering"
Qwen3-235B-A22B-Thinking (Yang et al., 2025a)	JJ CC VBG NN VB IN PRP . NNP	"complex or challenging question" "think about it . Well"
GPT-4o (OpenAI, 2024a)	VBZ : JJ NNP NNP CD NNP NNP NNP ,	"is : \ [\frac" "32c \ ] So ,"
GLM-4.6 (GLM-4.5 Team et al., 2025)	NN : ``) . RB JJ	"Wait : ``" ") . Consequently *"
Mixtral-8x7B-Instruct (Jiang et al., 2024)	. CD . RB , . RB VBZ DT NN	". 4 . Similarly ," ". Here 's the reasoning"
Llama-3-70B-Instruct (Grattafiori et al., 2024)	VBP PRP VBZ JJ . . PRP VBP PRP VBZ	"hope it is correct ." ". I hope it is"
GPT-3.5-turbo (Brown et al., 2020)	PRP VBP IN $ IN $ CD $	"we note that $" "Since $ 1111111100 $"
Qwen3-235B-A22B (Yang et al., 2025a)	# # # NNP CD . : : # #	"# # # Step 1" ". -- - # #"
Qwen3-32B (Yang et al., 2025a)	$ $: : # # # NNP	"$ $ --" "- # # # Key"

A Qualitative Examples of the PoS n-gram Signal

Signature Construction A model's signature comprises PoS n-grams it uses more often than its peers. We process the first 2,000 characters using NLTK Penn Treebank tags (Bird & Loper, 2004; Marcus et al., 1993), retain markup, and count 3–5-grams. Distinctiveness is $\ln\big((p_{kj}+\epsilon)/(\bar{p}_{-k,j}+\epsilon)\big)$, where $\epsilon = 10^{-5}$, p_{kj} is n-gram j's share in model k's profile, and $\bar{p}_{-k,j}$ is its mean share among

Table 4: **Candidate signatures (s1).** The two highest-ranked non-overlapping PoS n-grams of each candidate against the mean of the other eighteen, each with one matching span.

Candidate	Signature	Example
GPT-OSS-120B (OpenAI, 2025a)	. RB JJ NN) . NNP NN	". So -a cos" "} . Define c"
Gemma-3-27B-it (Gemma Team et al., 2025)	$. VB $ $. RB $	"$. Let $" "$. Then $"
Llama-3.3-70B-Instruct (Grattafiori et al., 2024)	. # # NNP CD CD : VB DT NN	". # # Step 2" "8 : Use the fact"
Qwen-3-8B (Yang et al., 2025a)	. NNP , VB PRP . VB PRP VB .	". First , let me" "? Let me think ."
Claude-Sonnet-3.5 (Anthropic, 2024)	. CD)) $ JJ $	". 1)" "} $ * $"
Claude-Opus-4.5 (Anthropic, 2025c)	CD $: $ CD $ VB PRP	"1 $: $ 10^1" "$ Let me"
Claude-Opus-4.6 (Anthropic, 2026b)	. RB PRP VBP VBG . # # VBG	". Now I 'm looking" ". # # Setting"
DeepSeek-R1 (Guo et al., 2025)	NNP , RB PRP VBP , VB PRP VB TO	"Alright , so I have" ", let me try to"
QwQ-32B-Preview (Qwen Team, 2024)	RB PRP VBP VBN DT VBP VBN DT NN RB	"So I 've got this" "'ve got this problem here"
o1 (OpenAI, 2024b)	PRP VBP JJ VBG NNP PRP VBP VBG	"I ' m considering" "* I 'm thinking"
o3 (OpenAI, 2025b)	NNP NNP PRP VBP TO VBP NN PRP VBP	"* * I need to" "* * I need"
Qwen3-235B-A22B-Thinking (Yang et al., 2025a)	DT VBZ DT JJ CC CC VBG NN , CC	"This is a complex or" "or challenging question , and"
GPT-4o (OpenAI, 2024a)	: JJ NNP NNP (, CD) NN)	": \ [\cos (" ", 0) \ )"
GLM-4.6 (GLM-4.5 Team et al., 2025)	. # # # . IN PRP VBP TO	". # # #" ". So we need to"
Mixtral-8x7B-Instruct (Jiang et al., 2024)	PRP VBP $ $ VB (NN NN)	"we have $ $" "\end { align * }"
Llama-3-70B-Instruct (Grattafiori et al., 2024)	VBP PRP VBZ JJ . . PRP VBP PRP VBZ	"hope it is correct ." ". I hope it is"
GPT-3.5-turbo (Brown et al., 2020)	$. NNP , DT JJ (CD) $	"$. Thus , the" "\boxed { 10800 } $"
Qwen3-235B-A22B (Yang et al., 2025a)	: # # # NNP . : : # #	"- # # # Step" ". -- - # #"
Qwen3-32B (Yang et al., 2025a)	VB PRP VB NN CD . VB PRP	"Let me place point" "2x . Let me"

the other models. We retain patterns appearing in at least 20 of 200 responses and report the top ten per probe set. These unfiltered lists are descriptive and do not enter SCOUT's score.

Candidate Signatures Tables 3 and 4 show two non-overlapping patterns from each list. The shared-prompt examples below use the four controlled candidates. Table 3 captures recurring forms: step headings (Llama-3.3-70B-Instruct, Qwen3-235B-A22B), first-person reasoning (Qwen-

3-8B, DeepSeek-R1, QwQ-32B-Preview), omitted subjects and articles (GPT-OSS-120B), inline-math definitions (Gemma-3-27B-it), and descriptions of ongoing reasoning (o1, o3). Repeated phrases also matter: Qwen3-235B-A22B-Thinking repeats one opening sentence in 126/200 OMI and 145/200 s1 responses; Llama-3-70B-Instruct uses "I hope it is correct" in 140/200 and 117/200. The candidate responses contain no GPT-OSS channel labels or Qwen `<think>` tags within the analyzed prefix. All ten leading patterns for Llama-3.3 and Qwen occur in variants of "## Step k:" and "let me," respectively, so each model uses a single highlight color below.

Same-Prompt Response Examples For each prompt q_i, we compute the mean pairwise Jensen–Shannon divergence among the four candidates' normalized signature vectors, restricted to patterns appearing at least five times across their responses. We show high-contrast examples from OMI and s1 that require no specialized background. All four candidate-versus-rest contrasts are positive on $161/200$ OMI prompts and $140/200$ s1 prompts, so the examples illustrate a common direction rather than typical contrast magnitude.

Line breaks are adjusted for readability. Wording and markup are otherwise preserved, with quotation marks typeset in the manuscript font. Leading or trailing period and comma tokens are left unhighlighted. † marks the end of the first $2{,}000$ characters of the original response.

OMI Example (prompt 140)

Ava has 18 vases with nine flowers in each vase. If 1/3 of them have 3 wilted flowers each, how many flowers are still fresh?

OMI / GPT-OSS-120B

. NNP JJ e.g. ". Need third"
Related patterns: NN . NNP NN, . NNP JJ NN,) NN . NNP

RB NN : e.g. "Thus answer :"
Related patterns: . RB NN, . RB NN :, . RB JJ NN, NN . RB JJ, . NN :

We need to parse: Ava has 18 vases with nine flowers in each vase. So total flowers = 18 * 9 = 162. If 1/3 of them have 3 wilted flowers each, "them" likely refers to vases? It says "1/3 of them have 3 wilted flowers each". Could be 1/3 of the vases have 3 wilted flowers each. So number of vases with wilted flowers = 1/3 * 18 = 6 vases. Each of those 6 vases has 3 wilted flowers. So total wilted flowers = 6 * 3 = 18 wilted flowers. All other flowers are fresh. Total flowers = 162, wilted = 18, fresh = 144. Thus answer: 144 fresh flowers. Check alternative interpretation: maybe 1/3 of the flowers have wilted - but phrase "them have 3 wilted flowers each" suggests vases. So answer 144. Thus final answer: 144 fresh flowers.First, find the total number of flowers:
\[
18\text{ vases}\times 9\text{ flowers per vase}=162\text{ flowers}
\]
Next, determine how many vases contain wilted flowers. One-third of the vases are affected:
\[
\frac13 \times 18 = 6\text{ vases}
\]
Each of those 6 vases has 3 wilted flowers, so the total number of wilted flowers is
\[
6\text{ vases}\times 3\text{ wilted flowers per vase}=18\text{ wilted flowers}
\]
Finally, subtract the wilted flowers from the total to get the fresh flowers:
\[
162\text{ total} - 18\text{ wilted}=144\text{ fresh flowers}
\]
\[

OMI / GPT-OSS-120B (continued)

\boxed{144}
\]

OMI / Llama-3.3-70B-Instruct

NNP CD e.g. "# # Step 1"
Related patterns: # # NNP CD :, # NNP CD, # NNP CD :, . # # NNP CD, . # #, . # # NNP, NNP CD : VB, # NNP CD : VB, NNP CD : VB DT

Step 1: Calculate the total number of flowers in all the vases.
Ava has 18 vases with 9 flowers in each vase. The total number of flowers is 18 * 9 = 162.
Step 2: Determine the number of vases with wilted flowers.
1/3 of the vases have 3 wilted flowers each. Since there are 18 vases, 1/3 of them is (1/3) * 18 = 6 vases.
Step 3: Calculate the total number of wilted flowers.
Each of the 6 vases has 3 wilted flowers. So, the total number of wilted flowers is 6 * 3 = 18.
Step 4: Calculate the number of fresh flowers.
The total number of flowers is 162, and the number of wilted flowers is 18. Therefore, the number of fresh flowers is 162 - 18 = 144.
The final answer is: $\boxed{144}$

OMI / Qwen-3-8B

VB PRP VB e.g. "Let me start"
Related patterns: . VB PRP VB, . VB PRP, , VB PRP VB, , VB PRP, NNP , VB PRP VB, NNP , VB PRP, VB PRP VB DT, . NNP , VB PRP, VB PRP VB IN

Okay, let me try to figure out this problem. So Ava has 18 vases, and each vase has 9 flowers. The question is asking how many flowers are still fresh, given that 1/3 of them have 3 wilted flowers each. Hmm, let me break this down step by step. First, let me find out the total number of flowers Ava has. If there are 18 vases and each vase has 9 flowers, then the total number of flowers is 18 multiplied by 9. Let me calculate that. 18 times 9... Well, 10 times 9 is 90, and 8 times 9 is 72, so adding those together, 90 + 72 is 162. So there are 162 flowers in total. Now, the problem says that 1/3 of them have 3 wilted flowers each. Wait, does that mean 1/3 of the vases or 1/3 of the flowers? Hmm, the wording is a bit ambiguous. Let me read it again: "1/3 of them have 3 wilted flowers each." The "them" here probably refers to the vases, since the previous sentence mentions vases. So it's likely that 1/3 of the vases have 3 wilted flowers each. Let me check that assumption. If it's 1/3 of the vases, then 1/3 of 18 vases is 6 vases. Each of those 6 vases has 3 wilted flowers. So, the number of wilted flowers would be 6 vases multiplied by 3 wilted flowers per vase, which is 18 wilted flowers. Therefore, the number of fresh flowers would be total flowers minus wilted flowers, which is 162 - 18 = 144. But let me make sure that's correct. Alternatively, if "1/3 of them" refers to 1/3 of the flowers, then it would be different. Let me consider that possibility as well. If 1/3 of the flowers are wilted, then 1/3 of 162 is 54 wilted flowers. Then the fresh flowers would be 162 - 54 = 108. But which interpretation is correct? Looking back at the problem: "Ava has 18 vases with nine flowers in each vase. If 1/3 of them have 3 wilted flowers each..." The "them" is likely referring to the vases because the previous sentence is about vases. So "them" would be the vases. Therefore, 1/3 of the vases have 3 wilted flowers each. So that would be 6 vases with 3 wilted flowers each.† Therefore, total wilted flowers are 6*3=18. So total fresh flowers are 162 - 18 = 144. That seems to make sense. But let me check again. If it was 1/3 of the flowers, then the problem might have said "1/3 of the flowers" instead of "1/3 of them."

OMI / Qwen-3-8B (continued)

Since “them” is used, which refers back to the vases. So I think the first interpretation is correct. So the answer would be 144. Wait, but let me think again. Maybe the problem is that each of those vases has 3 wilted flowers, meaning that in each of those vases, 3 flowers are wilted, so the rest are fresh. So for each of those 6 vases, there are 9 - 3 = 6 fresh flowers. Therefore, total fresh flowers would be (number of vases without wilted flowers) * 9 + (number of vases with wilted flowers) * (9 - 3). Let me calculate that.
Number of vases without wilted flowers is 18 - 6 = 12 vases. Each of those has 9 fresh flowers, so 12 * 9 = 108. Then the 6 vases with wilted flowers have 6 fresh flowers each, so 6 * 6 = 36. Then total fresh flowers are 108 + 36 = 144. So that’s the same as before. So that’s consistent. Alternatively, if I had done 162 total flowers minus 18 wilted flowers, that’s also 144. So that seems to check out. But let me just make sure that I didn’t misinterpret the problem. The problem says “1/3 of them have 3 wilted flowers each.” So “them” is the vases. So 1/3 of the vases, which is 6 vases, each have 3 wilted flowers. So that’s 6*3=18 wilted flowers. So total fresh is 162-18=144. So that seems correct. Alternatively, if someone thought that “1/3 of them” refers to 1/3 of the flowers, then it would be 54 wilted flowers. But that would mean that 54 flowers are wilted, which is 54/162 = 1/3. But the problem says “have 3 wilted flowers each.” If it was 1/3 of the flowers, then it would be 54 flowers, but how does that relate to 3 wilted flowers each? Maybe that’s a different interpretation. Let me think. If “1/3 of them have 3 wilted flowers each,” then maybe “them” refers to the flowers? But that would be odd. Because if “them” refers to flowers, then 1/3 of the flowers (which is 54) have 3 wilted flowers each. But that would mean that each of those 54 flowers has 3 wilted flowers? That doesn’t make sense. Because a flower can’t have 3 wilted flowers. So that interpretation is invalid. Therefore, the correct interpretation must be that “them” refers to the vases. Therefore, 1/3 of the vases have 3 wilted flowers each. Therefore, the answer is 144.
But let me check again. If “1/3 of them have 3 wilted flowers each,” then “them” could be ambiguous. But given the context, since the previous sentence is about vases, it’s more likely referring to the vases. Also, if it was referring to flowers, the sentence would probably say “1/3 of the flowers have 3 wilted flowers each,” which would be confusing. So the answer is 144. Alternatively, maybe the problem is that in each of the vases that have wilted flowers, 3 flowers are wilted. So, for each of those vases, 3 are wilted, so 9 - 3 = 6 are fresh. So total fresh flowers would be (18 vases - 6 vases with wilted flowers) * 9 + 6 vases * 6 flowers. Which is 12*9 + 6*6 = 108 + 36 = 144. Same answer. So that seems to confirm. Alternatively, if someone thought that 1/3 of the flowers are wilted, and each of those wilted flowers is in a vase with 3 wilted flowers... Wait, that seems more complicated. But the problem says “1/3 of them have 3 wilted flowers each.” So maybe “have 3 wilted flowers each” is referring to the vases. So each of those vases has 3 wilted flowers. So that’s 3 per vase. So that seems to be the correct interpretation. Therefore, I think the answer is 144. But let me check once more. Let me rephrase the problem. Total vases: 18. Each has 9 flowers. So total flowers: 18*9=162. 1/3 of them (vases) have 3 wilted flowers each. So 1/3 of 18 is 6 vases. Each of these 6 vases has 3 wilted flowers. So total wilted flowers: 6*3=18. Therefore, fresh flowers: 162-18=144. Yes, that seems right.
Alternatively, if someone thought that 1/3 of the flowers are wilted, but that would be 54 flowers. But the problem says “have 3 wilted flowers each.” So if you have 1/3 of the flowers, which is 54, but how does that relate to 3 wilted flowers each? Maybe that’s a different way of saying that each of those wilted flowers is in a group of 3? But that seems more complex and not how the problem is phrased. So I think the answer is 144. Therefore, I think the correct answer is 144.
Final Answer
The number of fresh flowers is \boxed{144}.
</think>
To determine how many flowers are still fresh, let’s break the problem down into logical steps.

Step 1: Total Number of Flowers
Ava has 18 vases, and each vase contains 9 flowers. So, the total number of flowers is:

OMI / Qwen-3-8B (continued)

$$
18 \times 9 = 162
$$

Step 2: Understanding the Wilted Flowers
The problem states

OMI / Gemma-3-27B-it

. RB $ e.g. ". Then $"
Related patterns: $. RB $

$ VB DT e.g. "$ be the"
Related patterns: $ VB DT NN, $ VB DT NN IN

. VB $ e.g. ". Let $"

Let N be the total number of vases Ava has, which is 18. Let F be the number of flowers in each vase, which is 9. The total number of flowers Ava has is $N \times F = 18 \times 9 = 162$. Let W be the fraction of vases that have wilted flowers, which is $\frac{1}{3}$. The number of vases with wilted flowers is $W \times N = \frac{1}{3} \times 18 = 6$. Each of these vases has 3 wilted flowers. The total number of wilted flowers is $6 \times 3 = 18$. The number of fresh flowers is the total number of flowers minus the number of wilted flowers.
Number of fresh flowers = Total number of flowers - Number of wilted flowers
Number of fresh flowers = $162 - 18 = 144$. Alternatively, we can calculate the number of fresh flowers in each vase with wilted flowers. Since each vase has 9 flowers and 3 are wilted, the number of fresh flowers in each vase with wilted flowers is $9 - 3 = 6$. The number of vases with wilted flowers is 6, so the total number of fresh flowers in these vases is $6 \times 6 = 36$. The number of vases without wilted flowers is $18 - 6 = 12$. Each of these vases has 9 fresh flowers, so the total number of fresh flowers in these vases is $12 \times 9 = 108$. The total number of fresh flowers is $36 + 108 = 144$.
Final Answer: The final answer is $\boxed{144}$

s1 Example (prompt 15)

Joanie takes a $6,000 loan to pay for her car. The annual interest rate on the loan is 12%. She makes no payments for 4 years, but has to pay back all the money she owes at the end of 4 years. How much more money will she owe if the interest compounds quarterly than if the interest compounds annually? Express your answer as a dollar value to the nearest cent.

s1 / GPT-OSS-120B

. RB JJ NN e.g. ". So -a cos"
Related patterns:) . RB JJ, . RB JJ, NN . RB JJ, . RB NN

. VB NNP VB e.g. ". Let ' s"
Related patterns: . VB NNP

. NNP NN e.g. "? Wait compute"
Related patterns: . NNP JJ, . NNP NN IN

We have a loan principal P=6000. Annual nominal interest rate r = 12% = 0.12. We need to compute amount owed after 4 years with compounding annually vs quarterly. Annual compounding: amount = P * (1 + r)^{4}. Quarterly compounding: rate per quarter = r/4 = 0.12/4 = 0.03. Number of quarters = 4 years *4 = 16. Amount = P * (1 + 0.03)^{16}. Compute both and find difference. Compute annual: (1.12)^4. Let's compute.
1.12^2 = 1.2544.
1.12^4 = (1.12^2)^2 = 1.2544^2.
Compute 1.2544^2:
1.2544 * 1.2544.
1.2544 * 1 = 1.2544
0.2544 *0.2544? No let's multiply precisely. Or use approximate: 1.2544^2 = ? Compute using calculator mental:
1.2544 * 1.2544.
Take 12544 *12544 scaled by 10^8 maybe. 12544 *12544: 12544*12544:
12544*12000 = 150,528,000
12544*544 = ?
12544*500 = 6,272,000
12544*40 = 501,760
12544*4 = 50,176
Sum = 6,823,936
Add to 150,528,000 = 157,351,936. Thus 12544*12544 = 157,351,936. Now scaling: Since 12544 = 1.2544*10^4? Actually 1.2544 = 12544/10000. So product = (12544^2)/(10000^2) = 157,351,936 / 100,000,000 = 1.57351936. So 1.12^4 = 1.57351936. Thus amount annual = 6000 * 1.57351936 = 9,441.11616. Now quarterly: (1.03)^16.
Compute (1.03)^16. We can compute using powers.
1.03^2 = 1.0609.
1.03^4 = (1.0609)^2.
Compute 1.0609^2:
1.0609 * 1.0609.
10609*10609 = ?
Let's compute: 10609*10609.
10609*10000 = 106,090,000
10609*600 = 6,365,400
10609*9 = 95,481
Sum = 112,550,881. Now scaling: because 1.0609 = 10609/10000. Square = 112,550,881/100,000,000 = 1.12550881. So 1.03^4 = 1.12550881. Now 1.03^8 = (1.03^4)^2 = (1.12550881)^2. Compute square: 1.12550881^2. Let's compute: 1.12550881 * 1.12550881. Convert to integer: 112550881 * 112550881 scaled by 10^16? Since 1.12550881 = 112550881 / 100,000,000. Square numerator: 112,550,881^2. That's big. Maybe use log approximations? Or compute stepwise using slight approximations.

s1 / GPT-OSS-120B (continued)

Alternatively compute 1.03^8 via incremental multiplication: 1.03^5 = 1.03^4 *1.03 = 1.12550881*1.03 = 1.1592740†743.
1.03^6 = *1.03 = 1.1940522965.
1.03^7 = *1.03 = 1.2298738654.
1.03^8 = *1.03 = 1.2667700814.
That seems easier: multiply sequentially. Let's continue: Actually we need up to 16. We have:
n value
1: 1.03
2: 1.0609
3: 1.092727
4: 1.12550881
5: 1.1592740743
6: 1.1940522965
7: 1.2298738654
8: 1.2667700814
9: *1.03 = 1.3047731838
10:*1.03 = 1.3439163793
11:*1.03 = 1.3842338707
12:*1.03 = 1.4257608868
13:*1.03 = 1.4685337124
14:*1.03 = 1.5125897238
15:*1.03 = 1.5579674155
16:*1.03 = 1.6047064380? Let's compute precisely. Let's do with more accurate rounding. Start from 1.12550881 we had accurate to 8 decimal; progress. We should compute with better precision perhaps using calculator style but okay. Let's recalc using high precision maybe with Python mental? Could approximate. We can compute (1.03)^16 directly using formula: (1+0.03)^16 ≈ e^{16 ln 1.03}. ln 1.03 ≈ 0.0295588. Multiply by 16 = 0.47294. e^{0.47294} ≈ 1.6045. So around 1.604. Our sequential multiplication gave 1.6047, close. Let's compute more precisely using incremental with more decimals. I'll keep more digits:
1.03^1 = 1.03
^2 = 1.0609
^3 = 1.0609*1.03 = 1.092727
^4 = 1.092727*1.03 = 1.12550881 (matches)
^5 = 1.12550881*1.03 = 1.1592740743
^6 = *1.03 = 1.1940522965 (since 1.1592740743*0.03=0.0347782222 add =>1.1940522965)
Let's compute precisely: 1.1592740743*0.03 = 0.034778222229, add = 1.194052296529. We'll keep 1.1940522965.
^7 = *1.03 => add 3%: 1.1940522965*0.03=0.0358215689, sum=1.2298738654 (like before). Keep maybe 1.2298738654.
^8 = *1.03 => 1.2298738654*0.03 = 0.03689621596, sum=1.26677008136.
^9 = *1.03 => add 0.03800310244 => 1.30477318380.
^10 = *1.03 => add 0.0391431955 => 1.34391637930.
^11 = *1.03 => add 0.0403174914 => 1.38423387070.
^12 = *1.03 => add 0.0415270161 => 1.42576088680.
^13 = *1.03 => add 0.0427728266 => 1.46853371340.
^14 = *1.03 => add 0.044055, compute: 1.4685337134*0.03=0.044055? Actually 1.4685337134*0.03 = 0.044055, precisely 0.044055, wait compute: 1.4685337134 * 0.03 = 0.044055, but need more digits: 1.4685337134 *0.03 = 0.044055... multiply 1.4685*3 = 4.4055 then divide by 100 =>0.044055. So sum = 1.5125887134? Wait add to 1.4685 = 1.5125887134.
But earlier we had 1.5125897238 slightly diff due to rounding. Let's compute more accurately: 1.4685337134 *0.03 = 0.044055, hmm 1.4685337134 *3 = 4.4056011402 divide 100 =0.044056011402. Add -> 1.512589724802. So value =1.5125897248.

s1 / GPT-OSS-120B (continued)

^15 = *1.03 => add 0.045377691744 => sum = 1.5579674165 (actually 1.5125897248 *0.03 = 0.045377691744; plus => 1.557967416544).
^16 = *1.03 => add 0.0467390224963 => sum = 1.604706438... compute: 1.557967416544*0.03 = 0.04673902249632; add => 1.60470643904032.
Thus (1.03)^16 ≈ 1.60470644.
Now amount quarterly = 6000 * 1.60470644 = 9

s1 / Llama-3.3-70B-Instruct

NNP CD e.g. "# # Step 1"
Related patterns: # NNP CD, # NNP CD :, # # NNP CD :, # # NNP, . # # NNP CD, . # # NNP, . # #, # NNP CD : VB, NNP CD : VB

Step 1: Calculate the amount owed if interest compounds annually
The formula for compound interest is $A = P(1 + r)^n$, where A is the amount of money accumulated after n years, including interest, P is the principal amount (initial amount of money), r is the annual interest rate (in decimal), and n is the number of years the money is invested or borrowed for. For annual compounding, $P = \$6,000$, $r = 0.12$, and $n = 4$. So, $A_{annual} = 6000(1 + 0.12)^4$.
Step 2: Calculate the amount owed if interest compounds quarterly
For quarterly compounding, the formula adjusts to $A = P(1 + \frac{r}{m})^{mt}$, where m is the number of times that interest is compounded per year and t is the time the money is invested or borrowed for, in years. Here, $P = \$6,000$, $r = 0.12$, $m = 4$ (since the interest is compounded quarterly), and $t = 4$. So, $A_{quarterly} = 6000(1 + \frac{0.12}{4})^{4*4}$.
Step 3: Compute A_{annual}
$A_{annual} = 6000(1 + 0.12)^4 = 6000(1.12)^4 = 6000 * 1.573520 = 9441.12$.
Step 4: Compute $A_{quarterly}$
$A_{quarterly} = 6000(1 + \frac{0.12}{4})^{4*4} = 6000(1 + 0.03)^{16} = 6000(1.03)^{16} = 6000 * 1.604706 = 9628.24$.
Step 5: Find the difference in the amounts owed
The difference in the amounts owed is $A_{quarterly} - A_{annual} = 9628.24 - 9441.12$.
Step 6: Calculate the final difference
$9628.24 - 9441.12 = 187.12$.
The final answer is: $\boxed{187.12}$

s1 / Qwen-3-8B

. VB PRP VB e.g. ". Let me try"
Related patterns: . VB PRP, VB PRP VB, , VB PRP VB, VB PRP VB IN, NNP , VB PRP, NNP , VB PRP VB, PRP VB IN, NN . VB PRP, . NNP , VB PRP

s1 / Qwen-3-8B (continued)

Okay, so Joanie took out a $6,000 loan with a 12% annual interest rate. She doesn't make any payments for 4 years and has to pay it all back at the end. The question is asking how much more she'll owe if the interest compounds quarterly compared to if it compounds annually. I need to find the difference between the two amounts. Alright, let me recall the formula for compound interest.
The formula for compound interest is A = P(1 + r/n)^(nt), where:
- A is the amount of money accumulated after n years, including interest.
- P is the principal amount (the initial amount of money).
- r is the annual interest rate (decimal).
- n is the number of times that interest is compounded per year.
- t is the time the money is invested or borrowed for, in years.
So, first, I need to calculate the amount she owes if the interest is compounded annually and then if it's compounded quarterly. Then subtract the two to find the difference.
Let me start with annual compounding. For annual compounding, n = 1. So plugging into the formula:
A_annual = 6000*(1 + 0.12/1)^(1*4)
Let me compute that. First, 0.12 divided by 1 is 0.12. Then 1 + 0.12 is 1.12. Then raise that to the power of 4. Let me calculate 1.12^4. Hmm, 1.12^2 is 1.2544. Then squaring that result for the fourth power. So 1.2544^2. Let me compute that. Let me do 1.2544 * 1.2544. First, multiply 1.25 * 1.25 = 1.5625. But since it's 1.2544, maybe I should do more precise calculation. Alternatively, maybe I can use logarithms or exponent rules, but maybe it's easier to just multiply step by step. Let me write it out: 1.2544 * 1.2544: Break it down as (1 + 0.25 + 0.0044)^2? Wait, maybe that's not helpful. Alternatively, use the standard multiplication method. Multiply 1.2544 by 1.2544: First, multiply 1.2544 by 1.2544:
Let me think of it as:
1.2544
x1.2544

Well, this might take some time. Alternatively, I can remember that 1.12^4 is a standard value. Let me check if I remember it or if I can calculate i†t step by step. Alternatively, use the formula for annual compounding:
After 1 year: 6000 * 1.12 = 6720
After 2 years: 6720 * 1.12. Let me compute that. 6720 * 1.12. 6720 * 1 = 6720, 6720 * 0.12 = 806.4. So total is 6720 + 806.4 = 7526.4
After 3 years: 7526.4 * 1.12. Let me compute that. 7526.4 * 1.12. Let me break it down:
First, 7526.4 * 1 = 7526.4
7526.4 * 0.12 = ?
Well, 7526.4 * 0.1 = 752.64
7526.4 * 0.02 = 150.528
So total is 752.64 + 150.528 = 903.168
Therefore, total after 3 years is 7526.4 + 903.168 = 8429.568
Then after 4 years: 8429.568 * 1.12. Let me compute that. Again, 8429.568 * 1.12. Break into 8429.568 * 1 + 8429.568 * 0.12.
First part is 8429.568
Second part: 8429.568 * 0.12. Let me calculate:
8429.568 * 0.1 = 842.9568
8429.568 * 0.02 = 168.59136
Adding those gives 842.9568 + 168.59136 = 1011.54816
Therefore, total amount after 4 years is 8429.568 + 1011.54816 = 9441.11616
So approximately $9,441.12 when compounded annually. Wait, but let me check if that's correct. Alternatively, maybe I can use the formula A = P*(1 + r)^t. Since it's annual compounding, n=1, so yeah, that's the same as A = 6000*(1.12)^4. Alternatively, maybe I can use a calculator for 1.12^4. Wait, but since I don't have a calculator here, maybe I can compute it more accurately. Wait, but I think my step-by-step calculation is correct.
Let me check again: First year: 6000 * 1.12 = 6720. Correct. Second year: 6720 * 1.12.

s1 / Qwen-3-8B (continued)

Let me compute 6720 * 1.12: Well, 6720 * 1.12 = 6720 + 6720 * 0.12 = 6720 + 806.4 = 7526.4. Correct. Third year: 7526.4 * 1.12. Let me compute 7526.4 * 1.12:
First, 7526.4 * 1.1 = 7526.4 + 752.64 = 8279.04
Then 7526.4 * 0.02 = 150.528
So total is 8279.04 + 150.528 = 8429.568. Correct. Fourth year: 8429.568 * 1.12.
Let me compute:
Again, 8429.568 * 1.1 = 8429.568 + 842.9568 = 9272.5248
Then 8429.568 * 0.02 = 168.59136
So total is 9272.5248 + 168.59136 = 9441.11616. So yes, $9,441.12 when rounded to the nearest cent. Okay, so that's the amount with annual compounding.
Now, moving on to quarterly compounding. Here, n = 4, since it's compounded quarterly.
So the formula becomes:
A_quarterly = 6000*(1 + 0.12/4)^(4*4) = 6000*(1 + 0.03)^16
Wait, 0.12 divided by 4 is 0.03. So 1 + 0.03 = 1.03. Then raised to the power of 16, since 4 years * 4 quarters per year = 16 periods. So I need to compute 1.03^16. Hmm, that might be a bit more involved. Let me see if I can calculate that step by step or find a way to approximate it. Alternatively, maybe I can use logarithms or remember some exponent values, but since I need to be precise, let me try to compute it step by step. Alternatively, maybe I can use the formula for compound interest step by step. Let me try that.
Alternatively, I can use the rule that (1 + r/n)^(nt). So 1.03^16. Let me compute that.
Let me note that 1.03^16 can be calculated as (1.03^4)^4. Maybe that's easier?
Let me compute 1.03^4 first. Compute 1.03^2 = 1.0609. Then 1.0609^2 = ?
Let me calculate 1.

s1 / Gemma-3-27B-it

. RB $ e.g. ". Then $"
Related patterns: $. RB $

. VB $ e.g. ". Let $"
Related patterns: $. VB $, VB $ VB, . VB $ VB, $ VB DT NN, $. VB $ VB, $ VB DT NN IN

Let P be the principal amount of the loan, which is $P = \$6000$. Let r be the annual interest rate, which is $r = 12\% = 0.12$. Let t be the number of years, which is $t = 4$.
If the interest compounds annually, the amount owed at the end of 4 years is
$$ A_1 = P(1+r)^t = 6000(1+0.12)^4 = 6000(1.12)^4 = 6000(1.57351936) = 9441.11616 \approx 9441.12. $$
If the interest compounds quarterly, the interest rate per quarter is $r_q = \frac{r}{4} = \frac{0.12}{4} = 0.03$. The number of quarters in 4 years is $n = 4t = 4(4) = 16$.
The amount owed at the end of 4 years is
$$ A_2 = P\left(1+\frac{r}{4}\right)^{4t} = 6000(1+0.03)^{16} = 6000(1.03)^{16} = 6000(1.6047064378) \approx 9628.2386268 \approx 9628.24. $$
The difference between the amount owed with quarterly compounding and the amount owed with annual compounding is
$$ A_2 - A_1 = 9628.24 - 9441.12 = 187.12. $$
The difference in the amount owed is
$$ A_2 - A_1 = 6000(1.03)^{16} - 6000(1.12)^4 = 6000((1.03)^{16} - (1.12)^4) $$
$$ = 6000(1.6047064378 - 1.57351936) = 6000(0.0311870778) \approx 187.12246668 \approx 187.12. $$
Final Answer: The final answer is $\boxed{187.12}$

Interpretation The lists depend on the candidate pool and signature size, and tagging errors can complicate their interpretation: sentence-initial "Compute b" receives . NNP NN. They describe recurring forms rather than establish attribution on their own.

Table 5: **Removing markup re-tags a signature rather than erasing it.** For each candidate in the controlled pool, we rank patterns against the mean of the other three using raw text and text with Markdown and math markup removed. The last column counts patterns common to the two top-ten lists.

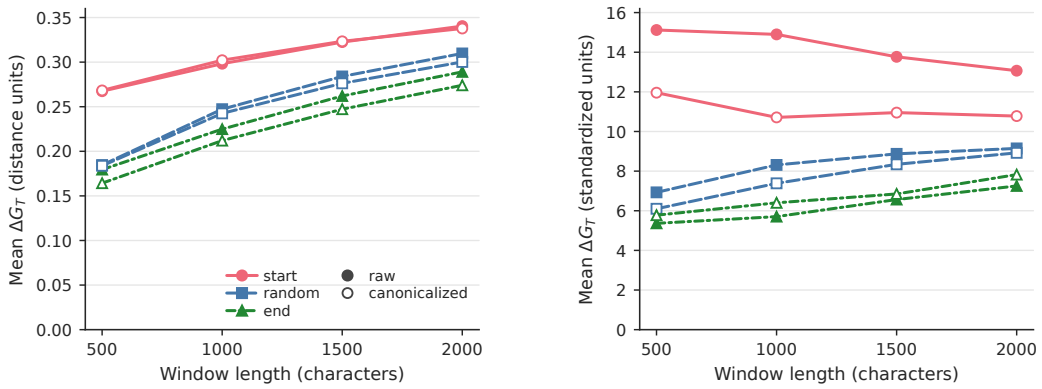
Candidate	Markup	Signature	Example	Shared
OMI probe set				
GPT-OSS-120B	kept	. NNP JJ	". Need third"	6/10
		RB NN :	"Thus answer :"	
	removed	RB NN :	"Thus answer :"	
		. NNP JJ	". Need third"	
Llama-3.3-70B-Instruct	kept	# # NNP CD	"# # Step 1"	0/10
		NNP CD : VB	"Step 1 : Define"	
	removed	NN CD : VB	"Step 1 : Define"	
Qwen-3-8B	kept	VB PRP VB	"Let me start"	9/10
	removed	, VB PRP VB	", let me plug"	
Gemma-3-27B-it	kept	. RB $	". Then $"	0/10
		$ $ $	"$ $ $"	
	removed	: DT JJ NN VBZ	": The final answer is"	
		. JJ NN : DT	". Final Answer : The"	
s1 probe set				
GPT-OSS-120B	kept	. RB JJ NN	". So -a cos"	5/10
		. VB NNP VB	". Let ' s"	
	removed	. NNP NN	"? Wait compute"	
		) . RB JJ	") . So -a"	
Llama-3.3-70B-Instruct	kept	# # NNP CD	"# # Step 1"	0/10
		NNP CD : VB	"Step 1 : Understand"	
	removed	CD : VB DT	"1 : Understand the"	
		. NN CD :	". Step 2 :"	
Qwen-3-8B	kept	. VB PRP VB	". Let me try"	9/10
		NNP , VB PRP	"First , let me"	
	removed	. VB PRP VB	". Let me try"	
		NNP , VB PRP	"First , let me"	
Gemma-3-27B-it	kept	. RB $	". Then $"	0/10
		. VB $	". Let $"	
	removed	VB NNP VB DT	"Let S be a"	
		CD (CD CD	"2 (1200 10^"	

B Representation Choice and Robustness

B.1 Markup and Window Controls

Markup Effects Table 5 shows that removing Markdown and math markup changes tag sequences without necessarily removing the underlying writing habits. Five or six of GPT-OSS-120B's top ten patterns remain unchanged, as do nine of Qwen-3-8B's. Although no exact top-ten pattern overlaps for Llama-3.3-70B-Instruct or Gemma-3-27B-it, the stripped responses retain step headings and definition-related phrases such as "Step 1: Define," "The final answer is," and "Let S be a." The factorial analysis below quantifies how these changes affect ΔG_T.

Text Length and Preprocessing The factorial analysis varies window length from 500 to 2,000 characters, position at the start, a random location, or the end, and preprocessing with raw or canonicalized text. Candidate-pool centering shifts G_T by the same constant for S and $S^{(0)}$, which cancels in ΔG_T. We therefore report D and C together. Figure 8 summarizes the trends, while Table 6 reports all 24 conditions under the default $\lambda = 0.15$.



(a) Profile D and Centered C

(b) SCOUT

Figure 8: **Teacher-directed margin changes across text windows and preprocessing.** Each point reports mean ΔG_T over nineteen controlled students under $\lambda = 0.15$. Filled and open markers denote raw and canonicalized text, respectively. Random-window curves average ten seeds. All nineteen students have positive ΔG_T in every condition. Panel **(a)** combines D and C because centering cancels in ΔG_T, while panel **(b)** applies candidate-specific scaling. The legend in **(a)** applies to both panels; their vertical scales are not comparable.

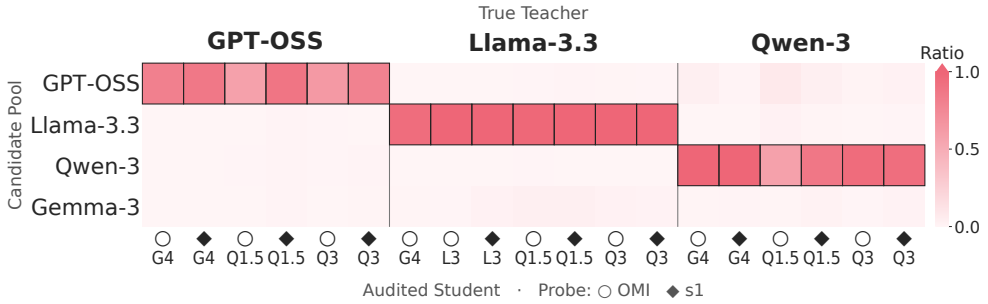
Table 6: **Trajectory-margin changes across text windows and preprocessing.** Each entry reports mean ΔG_T over nineteen controlled students. All nineteen have positive ΔG_T in every condition under D, C, and SCOUT. D and C are identical for ΔG_T, whereas SCOUT additionally applies candidate-specific scaling, so magnitudes should be compared only within each readout. Random-window means are averaged across ten seeds, and the $19/19$ result holds under the minimum count across seeds.

		Profile D and Centered C				SCOUT			
Preprocessing	Window	500	1000	1500	2000	500	1000	1500	2000
Raw	start	0.2674	0.2982	0.3225	0.3404	15.1212	14.8995	13.7677	13.0675
	random	0.1846	0.2472	0.2839	0.3097	6.9287	8.3123	8.8714	9.1461
	end	0.1795	0.2249	0.2620	0.2891	5.3654	5.7042	6.5654	7.2521
Canonicalized	start	0.2683	0.3021	0.3233	0.3377	11.9578	10.7116	10.9529	10.7779
	random	0.1838	0.2425	0.2762	0.3002	6.0996	7.3841	8.3427	8.9223
	end	0.1643	0.2120	0.2472	0.2740	5.7769	6.4008	6.8487	7.8247

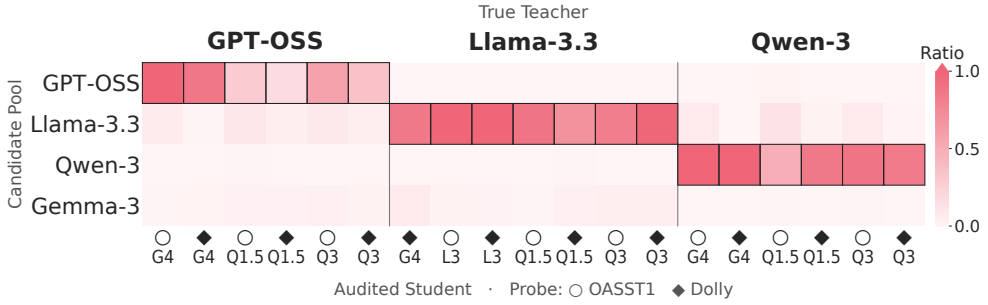
For D and C, mean ΔG_T increases with window length in all six preprocessing–position combinations. Under SCOUT, it increases for random and end windows, whereas both start-window series peak at 500 characters. Nevertheless, every student has positive ΔG_T in every condition. Relative to the first 2,000 raw characters, the first 500 canonicalized characters retain 78.8% of the gain under D and C, and 91.5% under SCOUT. Thus, teacher-associated evidence occurs throughout the response but is especially concentrated near its beginning.

B.2 Inheritance of Teacher Signatures

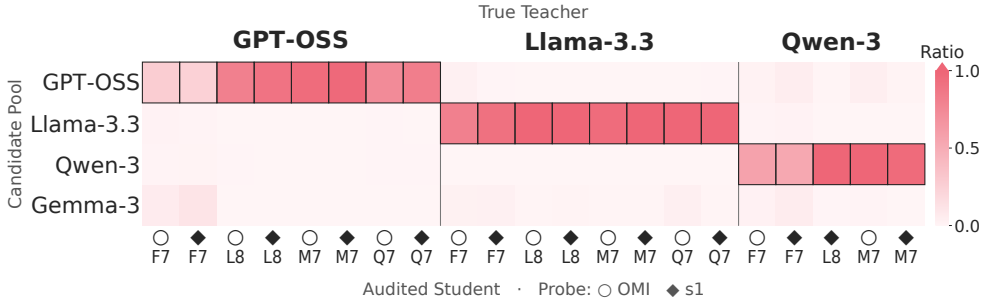
Signature Inheritance Figure 9 presents heatmaps of teacher-signature inheritance. Panels **(a)** and **(b)** repeat the 1.5–4B math and conversation results from Figure 5, while panel **(c)** adds the 7–8B math cohort. Each cell measures how strongly held-out student responses reproduce a candidate's top-ten signature after candidate-contrast filtering with $\lambda = 0.15$. The block-diagonal structure shows preferential reproduction of teacher-associated patterns.



(a) Math, 1.5–4B cohort



(b) Conversation, 1.5–4B cohort



(c) Math, 7–8B cohort

Figure 9: **Controlled students reproduce teacher-associated syntactic patterns.** Each cell shows the share of a candidate's top-ten PoS n-grams observed in held-out student responses, normalized by the corresponding share in the candidate's own responses. Signatures are selected after candidate-contrast filtering with $\lambda = 0.15$. Rows are candidates, columns are students grouped by true teacher, and black outlines mark each student's true teacher. Student labels G4, L3, Q1.5, Q3, F7, L8, M7, and Q7 denote Gemma-3-4B-PT, Llama-3.2-3B-Instruct, Qwen-2.5-1.5B, Qwen-2.5-3B, Falcon3-7B-Base, Llama-3.1-8B, Mistral-7B-v0.3, and Qwen-2.5-7B, respectively. Circles and diamonds distinguish the two probes used in each cohort.

Table 7: **Student signatures.** Controlled students grouped by source, against the students of the other two sources on the same base models, with format scaffolding removed.

Cohort	Source	Probe	Signature	Example
1.5–4B	GPT-OSS-120B	OMI	. NNP NN	". Compute b"
			) . RB JJ	") . Similarly median"
		s1	. RB JJ	". Similarly \"
			. NNP NN	". Compute product"
	Llama-3.3-70B-Instruct	OMI	NNP CD : VB DT	"Step 2 : Recall the"
			. # # NNP CD	". # # Step 2"
		s1	. # # NNP CD	". # # Step 2"
			# NNP CD : VB	"# Step 1 : Understand"
	Qwen-3-8B	OMI	. VB PRP VB	". Let me try"
			NNP , VB PRP	"First , let me"
		s1	. VB PRP VB	"? Let me write"
			NNP , VB PRP	"Hmm , let me"
7–8B	GPT-OSS-120B	OMI	) . RB JJ	") . Thus wˆ-6"
			. NNP NN	". Done computation"
		s1	) . RB JJ	"} . Thus upper"
			. NNP NN	"? Simplify orig"
	Llama-3.3-70B-Instruct	OMI	. # # NNP	". # # Step"
			# NNP CD : VB	"# Step 3 : Apply"
		s1	. # # NNP CD	". # # Step 2"
			# NNP CD : VB	"# Step 1 : Consider"
	Qwen-3-8B	OMI	. VB PRP VB	". Let me recall"
			NNP , VB PRP	"First , let me"
		s1	. VB PRP VB	". Let me come"
			NNP , VB PRP	"Okay , let me"

Training-Scaffold Removal Table 7 lists representative student-side patterns after removing training-format markers, specifically GPT-OSS channel labels and Qwen `<think>` tags. Teacher-associated forms recur across both probes and model scales, indicating that the inheritance pattern is not solely driven by copied scaffolding.

Signature-Size Sensitivity Figure 10 tests whether the true teacher remains top-ranked as the signature size and candidate pool vary. The ranking is stable around the default $k = 10$, but broader lists eventually include patterns shared across models.

B.3 Candidate-Contrast Sensitivity

The filtering fraction λ controls how many low-contrast PoS n-grams are removed before profile distances are computed. This representation-stage filter is separate from the candidate-pool calibration. We examine sensitivity to λ across controlled cohorts and public descendants after subsequent training. The analysis characterizes a stable region rather than selecting an optimal value.

Controlled Cohorts Table 8 shows consistent behavior at $\lambda = 0.10$ and 0.15. Both yield no false acceptances under either threshold-transfer protocol in any cohort while retaining high detection. Performance is less consistent outside these values, particularly in the conversation cohort and at $\lambda = 0.25$. The same pattern in the additional 7–8B cohort indicates that it is not confined to smaller students. All primary experiments use the fixed value $\lambda = 0.15$, one point in this stable region.

After Subsequent Training Table 9 shows a consistent pattern after subsequent training. SCOUT assigns all twelve descendants to DeepSeek-R1 across the evaluated grid on both math probes. On the conversation probes, OASST1 and Dolly simultaneously reach $12/12$ at $\lambda = 0.10$ and 0.15. In

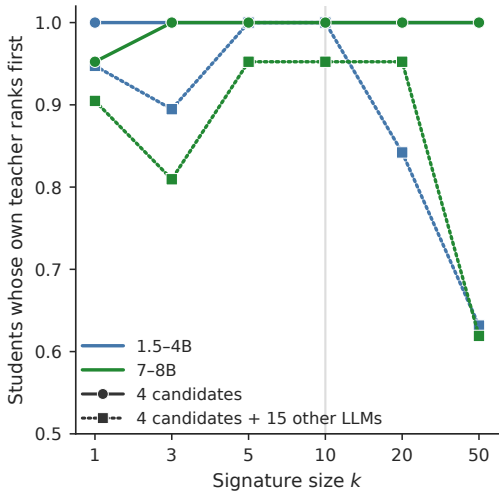


Figure 10: **Sensitivity to signature size and candidate-pool composition.** The fraction of controlled students whose true source ranks first as the number of signature PoS n-grams varies. Solid lines with circles use the four-candidate pool, whereas dotted lines with squares use the nineteen-candidate pool. The vertical line marks the default choice, $k = 10$.

Table 8: **Sensitivity to λ in the controlled cohorts.** Det. counts students correctly attributed on both probes with the teacher present. Each ordered pair $(\mathrm{FP}(S), \mathrm{FP}(T))$ gives the numbers of false acceptances after teacher removal under student-backbone and teacher holdout, respectively. Bold headers mark the only values yielding no false acceptances under either protocol in all three cohorts.

Cohort	Measure	$\lambda = 0$	0.05	**0.10**	**0.15**	0.25	0.50	0.75
Math 1.5–4B (19)	Det.	19/19	15/19	18/19	18/19	18/19	17/19	17/19
	$(\mathrm{FP}(S), \mathrm{FP}(T))$	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(13, 19)	(0, 0)	(0, 0)
Conversation (19)	Det.	8/19	12/19	15/19	15/19	15/19	15/19	16/19
	$(\mathrm{FP}(S), \mathrm{FP}(T))$	(2, 2)	(2, 0)	(0, 0)	(0, 0)	(0, 4)	(2, 1)	(2, 0)
Additional scale cohort								
Math 7–8B (21)	Det.	19/21	15/21	19/21	18/21	19/21	18/21	18/21
	$(\mathrm{FP}(S), \mathrm{FP}(T))$	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(13, 21)	(0, 0)	(0, 0)

every SCOUT cell with $12/12$ descendants, both distilled parents are also assigned to DeepSeek-R1 and both pre-distillation bases are correctly rejected as non-R1. Profile distance D does not recover the conversation descendants, indicating that contrast filtering alone does not explain SCOUT's performance. Together, the controlled and retrospective results identify 0.10–0.15 as a stable region across model scales, probe domains, and training stages.

B.4 Decoding and Response-Length Controls

This subsection tests whether the measured signal can be explained by differences in teacher-side decoding, candidate-side decoding, or total response length.

Common Teacher Decoding The controlled training responses were generated using each teacher's default decoding configuration. Qwen-3-8B uses $T = 0.6$, top-$p = 0.95$, and top-$k = 20$; Llama-3.3-70B uses $T = 0.6$ and top-$p = 0.9$; GPT-OSS-120B provides no sampling defaults and was served at $T = 1.0$. Because these differences may affect PoS patterns, we regenerate training responses from all three teachers using $T = 0.7$, top-$p = 0.95$, top-$k = 20$, and repetition penalty 1.0, then train the corresponding students with the original recipe.

Table 9: **Sensitivity to λ after subsequent training.** Each entry counts DeepSeek-R1 assignments among the twelve descendants in Section 5.2. Math and conversation probes use their corresponding ten-candidate pools. Profile distance D uses the same filtered representations without candidate-pool calibration. Bold headers mark the values highlighted by the controlled cohorts.

Method	Probe	$\lambda = 0$	0.05	**0.10**	**0.15**	0.25	0.50
SCOUT	Math / OMI	12/12	12/12	12/12	12/12	12/12	12/12
	Math / s1	12/12	12/12	12/12	12/12	12/12	12/12
	Conversation / OASST1	1/12	5/12	12/12	12/12	1/12	1/12
	Conversation / Dolly	3/12	12/12	12/12	12/12	12/12	1/12
Profile distance D	Math / OMI	11/12	12/12	12/12	12/12	12/12	12/12
	Math / s1	12/12	12/12	12/12	12/12	12/12	12/12
	Conversation / OASST1	1/12	1/12	2/12	2/12	2/12	1/12
	Conversation / Dolly	1/12	1/12	1/12	1/12	1/12	1/12

Table 10: **Source-signature formation under common teacher decoding.** Entries report the source-directed increment I_T on the held-out probe. Dashes denote combinations outside the retained nineteen-student cohort.

Backbone	Training set	GPT-OSS	Qwen-3	Llama-3.3
Gemma-3-4B-PT	OMI	0.087	0.169	—
	s1	0.146	0.157	0.191
Llama-3.2-3B-Instruct	OMI	—	—	0.192
	s1	—	—	0.228
Qwen2.5-1.5B	OMI	0.108	0.179	0.139
	s1	0.169	0.081	0.176
Qwen2.5-3B	OMI	0.108	0.203	0.168
	s1	0.175	0.234	0.191
Mean I_T		0.132	0.170	0.184
True source ranks first		6/6	6/6	7/7

For each candidate T_k, let Δ_k denote the reduction in PoS distance from the base to the student. We measure source-directed formation as

$$I_T = \Delta_T - \frac{1}{K-1} \sum_{k \neq T} \Delta_k,$$

where positive values indicate greater movement toward the true source than toward the remaining candidates.

Table 10 shows positive I_T for every student in the nineteen-student cohort. The largest candidate-wise distance reduction identifies the true source in all $19/19$ cases. The mean I_T is 0.163, compared with 0.167 for the same students under the original teacher configurations.

For exact inference, we permute teacher labels within five three-teacher blocks and one two-teacher block. These blocks cover 17 students; the other two Llama-3.2-3B-Instruct students have no within-block alternative. All 17 students have positive I_T and move most toward the true source, with a mean I_T of 0.158. The observed assignment uniquely maximizes the mean among all $(3!)^5 2! = 15{,}552$ assignments, giving an exact one-sided $p = 6.4 \times 10^{-5}$. Equalizing teacher decoding therefore does not remove source-signature formation.

Temperature and Candidate-Side Decoding We next measure the magnitude of the decoding effect. GPT-OSS training responses are generated at three temperatures while all other settings remain fixed. On the candidate side, we regenerate Gemma-3-27B-it, Llama-3.3-70B-Instruct, GPT-OSS-120B, and QwQ-32B-Preview under both their model defaults and the common configuration above. The other six candidates, including the true source DeepSeek-R1, remain fixed. We denote the true-source margin computed from uncalibrated profile distance D by $G_T^{(D)}$.

Table 11: **Effects of teacher- and candidate-side decoding.** The upper block reports $G_T^{(D)}$ for students trained on GPT-OSS responses at three temperatures. The middle block compares its mean across teachers under common decoding. In the lower block, *Released* uses the original candidate bank, while *Default* and *Common* replace four open-weight competitors with regenerated responses. Entries count correct source identifications in thirteen public suspect–probe cells using profile readouts with $\lambda = 0.15$. The s1.1-32B cell on s1 is excluded because its fine-tuning data contain the probe prompts. *Flips* counts verdict changes between default and common candidate decoding.

	Temperature			
Backbone	$T = 0.3$	$T = 0.7$	$T = 1.0$	Range
Gemma-3-4B-PT	0.0839	0.0865	0.0937	0.0098
Qwen2.5-1.5B	0.0602	0.0777	0.0697	0.0176
Qwen2.5-3B	0.0833	0.0988	0.0845	0.0155
Mean range				0.0143
	GPT-OSS	Qwen-3	Llama-3.3	Range
Mean $G_T^{(D)}$	0.0647	0.1400	0.2578	0.1931
Readout	Released	Default	Common	Flips
Profile distance D	13/13	13/13	13/13	0/13
+ Centering C	13/13	13/13	13/13	0/13
+ Scaling (SCOUT)	13/13	13/13	13/13	0/13

Table 12: **Response-length controls.** The upper block reports candidate-response statistics on OMI before and after the 2,000-character cap; entries are medians except *At cap*, which gives the percentage of responses reaching the cap. *Sentence chars* reports median sentence length within the retained window. The lower block compares three length-only readouts with SCOUT on the same nineteen controlled students. SCOUT uses $\lambda = 0.15$, and lower-block entries are percentages.

Candidate	Raw chars	Retained chars	At cap (%)	Sentence chars
GPT-OSS-120B	2,877	2,000	63.5	54.7
Qwen-3-8B	6,036	2,000	100.0	62.5
Llama-3.3-70B-Instruct	1,851	1,851	44.5	117.6
Gemma-3-27B-it	2,013	1,993	50.0	80.2
Max/min ratio	3.26	1.08	—	2.15

Readout	GPT-OSS	Qwen-3	Llama-3.3
Character count	100.0	0.0	14.3
Token count	66.7	83.3	0.0
Line count	50.0	50.0	42.9
SCOUT (Ours)	100.0	100.0	100.0

The mean temperature-induced range in $G_T^{(D)}$ is 0.0143, compared with a 0.1931 range between teachers. Changing candidate-side decoding alters none of the 39 verdicts across the three profile readouts. The tested training-side temperature variation is therefore substantially smaller than the source variation, and changing the four regenerable competitors does not alter attribution.

Response-Length Controls At audit time, SCOUT retains only the first 2,000 characters of every student and candidate response and L_2-normalizes the resulting count vector. Table 12 measures the remaining length variation and tests whether simple length statistics recover the source. For each prompt, a length-only readout scores candidate T_k by the absolute difference between the student's response statistic and the median statistic over that candidate's 200 responses; the released evaluator then aggregates these scores across prompts.

The character cap reduces the ratio between the longest and shortest candidate medians from 3.26 to 1.08. Each length-only readout succeeds for some sources but fails for others, whereas SCOUT

identifies every source group. Therefore, total length does not explain the controlled result, although sentence structure within the retained window may still contribute to the PoS representation.

Together, these controls show that neither the tested decoding variation nor total retained length alone accounts for the attribution results.

C Properties and Limits of the SCOUT Readout

This appendix establishes four properties of SCOUT. It identifies what candidate-pool calibration removes, when attribution remains stable under prompt subsampling, when candidate-only calibration cannot remove pair-specific affinity, and what current outputs cannot reveal about training history. The calibration identities condition on each supplied candidate pool, while the subsampling analysis fixes the candidate responses, profiles, null statistics, and scales. Randomness in the latter enters only through subsets drawn from a fixed prompt bank. These results characterize the readout rather than the causal effects of distillation or post-training. Then, we connect the final limit to source-switching experiments that examine which source remains readable after later SFT.

C.1 Candidate-Relative Calibration

Proposition 1 (Exact finite-pool calibration)**.** *For any evaluated model A, we have*

$$C(A, T_k) = D(A, T_k) - \frac{1}{K-1} \sum_{j \neq k} D(T_j, T_k). \tag{12}$$

Moreover, assume $\sigma(T_k) > 0$ and suppose every distance to T_k used in its calibrated score receives the same candidate-specific additive shift b_k:

$$\delta_i'(M, T_k) = \delta_i(M, T_k) + b_k$$

for every $M \in \{A\} \cup (\mathcal{T} \setminus \{T_k\})$ and prompt i. Then,

$$C'(A, T_k) = C(A, T_k), \qquad Z'(A, T_k) = Z(A, T_k). \tag{13}$$

Proof. Exchanging the order of summation gives

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^{N} \mu_i(T_k) &= \frac{1}{N(K-1)} \sum_{i=1}^{N} \sum_{j \neq k} \delta_i(T_j, T_k) \\ &= \frac{1}{K-1} \sum_{j \neq k} D(T_j, T_k). \end{aligned}$$

Substitution into the definition of C yields Eq. 12. Under the additive shift, $\mu_i'(T_k) = \mu_i(T_k) + b_k$, so every centered distance is unchanged. The standard deviation is also unchanged because all distances entering it receive the same shift. Hence both C and Z remain unchanged. □

Thus, centering removes an offset shared by the evaluated model and all pool members relative to T_k. It prevents T_k from being favored solely because all models are uniformly close to it. The null is computed without the evaluated model. Therefore, a low $Z(A, T_k)$ indicates closeness relative to the supplied pool but does not by itself establish distillation.

Effect of pool composition Eq. 12 also shows how the supplied pool affects centering. For $K \geq 3$, holding the representation, profiles, and distances fixed, removing a candidate T_r with $r \neq k$ changes the centered score by

$$C^{(-r)}(A, T_k) - C(A, T_k) = \frac{1}{N(K-2)} \sum_{i=1}^{N} \left[\delta_i(T_r, T_k) - \mu_i(T_k) \right]. \tag{14}$$

Therefore, including a pool member that is unusually close to T_k makes T_k's centered score less favorable. This identity isolates the centering term because removing a candidate also changes $\sigma(T_k)$. In our implementation, changing the pool may also alter the PoS n-gram vocabulary and candidate-contrast mask w_λ, thereby changing the distances themselves.

C.2 Finite-Prompt Stability

For a fixed audited model S, define the standardized per-prompt score

$$z_{ik} = \frac{\delta_i(S, T_k) - \mu_i(T_k)}{\sigma(T_k)}. \tag{15}$$

For a subset I of B audit prompts, let

$$Z_I(S, T_k) = \frac{1}{B} \sum_{i \in I} z_{ik}. \tag{16}$$

Proposition 2 (Stability under prompt subsampling). *Assume $\sigma(T_k) > 0$ for every candidate. Let $T_{k^\star}$ be the true source and suppose it uniquely minimizes the full-bank score. For each $k \neq k^\star$, define*

$$g_{ik} = z_{ik} - z_{ik^\star}, \qquad \gamma_k = \frac{1}{N} \sum_{i=1}^{N} g_{ik} = Z(S, T_k) - Z(S, T_{k^\star}) > 0,$$

with range width

$$L_k = \max_i g_{ik} - \min_i g_{ik} > 0.$$

If I is sampled uniformly without replacement from the N audit prompts, with $1 \leq B \leq N$, then

$$\Pr\left(\arg\min_k Z_I(S, T_k) \neq k^\star \right) \leq \sum_{k \neq k^\star} \exp\left(-\frac{2B\gamma_k^2}{L_k^2} \right). \tag{17}$$

Consequently, with $\gamma = \min_{k \neq k^\star} \gamma_k$ and $L = \max_{k \neq k^\star} L_k$,

$$B \geq \frac{L^2}{2\gamma^2} \log \frac{K-1}{\varepsilon} \tag{18}$$

is sufficient for subsampling error at most ε.

Proof. A competitor T_k can defeat $T_{k^\star}$ only if

$$Z_I(S, T_k) - Z_I(S, T_{k^\star}) = \frac{1}{B} \sum_{i \in I} g_{ik} \leq 0.$$

Applying Hoeffding's inequality for sampling without replacement (Hoeffding, 1963) to each competitor gives

$$\Pr\left(\frac{1}{B} \sum_{i \in I} g_{ik} \leq 0 \right) \leq \exp\left(-\frac{2B\gamma_k^2}{L_k^2} \right).$$

A union bound over the $K-1$ competitors yields Eq. 17. Bounding every summand using γ and L and solving for B gives Eq. 18. □

This result is a stability guarantee rather than an unconditional attribution guarantee. It assumes that the true source wins on the full prompt bank and bounds the probability that prompt subsampling changes that ranking. The required budget is governed by the hardest competitor rather than average separation across the pool. Requiring all J audited models to retain the correct attribution adds a union bound over models and replaces $K-1$ in Eq. 18 with $J(K-1)$, with γ and L taken over all model and competitor pairs.

The bound applies directly to mean-score attribution, for which attribution is determined by the audited-model average in Eq. 16. It does not directly cover the sorted-score voting component included in the original-evaluator rows of Tables 13 and 14. In both cases, candidate responses, profiles, contrast masks, null statistics, and scales remain fixed while only the audited-model responses are subsampled.

Table 13: **Prompt-budget curves on math probes.** Each value for $B < 200$ is the fraction of 2,000 prompt subsets on which all twelve descendants are assigned to R1. Under the original evaluator, each descendant is correct when either mean-score attribution or sorted-score voting assigns it to R1, and a subset succeeds only when all twelve descendants are correct. Mean-score attribution uses the lowest average candidate score alone. At $B = 200$, the complete prompt bank is evaluated once. $n^\star$ is the first evaluated budget reaching 0.95. Dashes indicate that this level is not reached.

Method	Aggregation	5	10	20	50	100	110	120	130	140	150	160	175	190	200	$n^\star$
OMI																
Reference-based																
DistillDetect	Original evaluator	0.9885	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	5
DistillDetect	Mean-score attribution	0.9585	0.9905	0.9995	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	5
Output-only																
PoS Templates	Original evaluator	0.0115	0.0095	0.0025	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
PoS Templates	Mean-score attribution	0.0060	0.0060	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
LLM judge (Mistral)	Most frequent label	0.0045	0.0280	0.0860	0.2835	0.5495	0.5630	0.6085	0.6310	0.6640	0.6960	0.7060	0.7685	0.8875	1.0000	200
LLM judge (Solar)	Most frequent label	0.0315	0.1155	0.2525	0.6075	0.8900	0.9230	0.9485	0.9630	0.9825	0.9895	1.0000	1.0000	1.0000	1.0000	130
SCOUT (Ours)	Original evaluator	0.5400	0.7025	0.8665	0.9830	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	50
SCOUT (Ours)	Mean-score attribution	0.3975	0.5995	0.8160	0.9800	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	50
− Scaling (C)	Original evaluator	0.5180	0.6850	0.8525	0.9825	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	50
− Scaling (C)	Mean-score attribution	0.3920	0.5915	0.8040	0.9775	0.9995	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	50
− Centering (D)	Original evaluator	0.4360	0.4500	0.5730	0.6975	0.7650	0.7660	0.7940	0.8300	0.8290	0.8405	0.8795	0.9305	0.9930	1.0000	190
− Centering (D)	Mean-score attribution	0.2965	0.3245	0.4175	0.5485	0.6785	0.6800	0.7105	0.7630	0.7760	0.7930	0.8475	0.9150	0.9930	1.0000	190
s1																
Reference-based																
DistillDetect	Original evaluator	0.7270	0.7990	0.8520	0.9415	0.9950	0.9980	0.9985	0.9995	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	100
DistillDetect	Mean-score attribution	0.6525	0.6995	0.7560	0.8820	0.9825	0.9915	0.9970	0.9990	0.9995	1.0000	1.0000	1.0000	1.0000	1.0000	100
Output-only																
PoS Templates	Original evaluator	0.0320	0.0360	0.0325	0.0095	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
PoS Templates	Mean-score attribution	0.0185	0.0215	0.0185	0.0060	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
LLM judge (Mistral)	Most frequent label	0.0010	0.0050	0.0280	0.1235	0.2840	0.3275	0.3865	0.3770	0.3930	0.4175	0.4460	0.4285	0.4095	0.0000	—
LLM judge (Solar)	Most frequent label	0.0195	0.0505	0.1470	0.4705	0.8035	0.8275	0.8850	0.9025	0.9360	0.9620	0.9840	0.9975	1.0000	1.0000	150
SCOUT (Ours)	Original evaluator	0.7745	0.9340	0.9955	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	20
SCOUT (Ours)	Mean-score attribution	0.6680	0.8835	0.9780	0.9995	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	20
− Scaling (C)	Original evaluator	0.7635	0.9290	0.9925	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	20
− Scaling (C)	Mean-score attribution	0.6630	0.8810	0.9760	0.9995	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	20
− Centering (D)	Original evaluator	0.9155	0.9935	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	10
− Centering (D)	Mean-score attribution	0.8760	0.9875	0.9990	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	10

Prompt-budget resampling We evaluate the twelve descendants from Section 5.2. For each budget $B < 200$, we draw 2,000 uniformly random B-prompt subsets without replacement from the 200-prompt bank and apply each subset to all descendants. The full bank at $B = 200$ is evaluated once. For score-based methods, the original evaluator counts each descendant as correct when either mean-score attribution or sorted-score voting assigns it to R1, and a subset succeeds only when all twelve descendants are correct. We also report mean-score attribution alone because it corresponds directly to Proposition 2. Each LLM judge assigns every descendant using its unique most frequent prompt-level label, with ties yielding no verdict. Tables 13 and 14 report the resulting full curves and the first evaluated budget $n^\star$ at which the success rate reaches 0.95.

Across the four probes, the two score aggregations yield the same $n^\star$ except on Dolly, where SCOUT and its centered variant move from 10 prompts under the original evaluator to 20 under mean-score attribution. On OMI, SCOUT reaches the criterion at 50 prompts under both aggregations, earlier than the uncalibrated profile and output-only baselines. Raw profile distance reaches the criterion earlier on s1 but never reaches it on either conversation probe. On OASST1 and Dolly, centering separates SCOUT from every baseline, while candidate-specific scaling produces nearly identical prompt-budget curves. These results show that the conversation performance does not depend on the original evaluator combining two score aggregations.

Stability of attribution and abstention Let $\mathbf{z}$ and $\mathbf{z}'$ be two candidate-score vectors satisfying $\|\mathbf{z}' - \mathbf{z}\|_\infty \le \eta$, and let $H(\mathbf{z})$ be the gap between the smallest and second-smallest scores. Since both order statistics can change by at most η, $|H(\mathbf{z}') - H(\mathbf{z})| \le 2\eta$. The selected candidate is unchanged when $H(\mathbf{z}) > 2\eta$. For a fixed threshold τ, the accept–abstain decision is unchanged whenever $H(\mathbf{z}) \ge \tau + 2\eta$ or $H(\mathbf{z}) < \tau - 2\eta$. Therefore, the decision margin in Eq. 9 also measures robustness to score perturbations.

C.3 Limits of Candidate-Only Calibration

Candidate-pool calibration applies the same candidate-specific correction to every audited model. Such a correction fully removes an interaction when the remaining score differs from the source-

Table 14: **Prompt-budget curves on conversation probes.** Entries follow Table 13. Each value for $B < 200$ is the fraction of 2,000 prompt subsets on which all twelve descendants are assigned to R1. At $B = 200$, the complete prompt bank is evaluated once. Dashes indicate that the 0.95 criterion is not reached.

Method	Aggregation	5	10	20	50	100	110	120	130	140	150	160	175	190	200	$n^\star$
OASST1																
Reference-based																
DistillDetect	Original evaluator	0.0740	0.0275	0.0060	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
DistillDetect	Mean-score attribution	0.0540	0.0220	0.0035	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
Output-only																
PoS Templates	Original evaluator	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
PoS Templates	Mean-score attribution	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
LLM judge (Mistral)	Most frequent label	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
LLM judge (Solar)	Most frequent label	0.0025	0.0030	0.0005	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
SCOUT (Ours)	Original evaluator	0.6995	0.9225	0.9955	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	20
SCOUT (Ours)	Mean-score attribution	0.6265	0.8940	0.9930	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	20
− Scaling (C)	Original evaluator	0.6940	0.9190	0.9935	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	20
− Scaling (C)	Mean-score attribution	0.6105	0.8825	0.9880	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	20
− Centering (D)	Original evaluator	0.0065	0.0010	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
− Centering (D)	Mean-score attribution	0.0035	0.0010	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
Dolly																
Reference-based																
DistillDetect	Original evaluator	0.1750	0.1120	0.0510	0.0045	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
DistillDetect	Mean-score attribution	0.1415	0.1005	0.0420	0.0035	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
Output-only																
PoS Templates	Original evaluator	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
PoS Templates	Mean-score attribution	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
LLM judge (Mistral)	Most frequent label	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
LLM judge (Solar)	Most frequent label	0.0025	0.0055	0.0045	0.0015	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
SCOUT (Ours)	Original evaluator	0.8755	0.9805	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	10
SCOUT (Ours)	Mean-score attribution	0.8055	0.9495	0.9950	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	20
− Scaling (C)	Original evaluator	0.8780	0.9815	0.9995	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	10
− Scaling (C)	Mean-score attribution	0.8035	0.9490	0.9950	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	20
− Centering (D)	Original evaluator	0.0010	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—
− Centering (D)	Mean-score attribution	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	—

related component only by a term shared across candidates. The following result characterizes when this is possible.

Proposition 3 (Candidate-only correction)**.** *Suppose an audited-model–candidate distance admits the decomposition*

$$d(S,k) = t(S,k) + a(S,k) + b(k), \tag{19}$$

where $t(S,k)$ is the source-related component, $b(k)$ is a candidate-wide component, and $a(S,k)$ is an audited-model–candidate interaction. Fix a positive candidate-dependent scale $s(k)$. There exist a candidate-only correction $c(k)$ and a function $h(S)$ such that

$$\frac{d(S,k) - c(k)}{s(k)} = \frac{t(S,k)}{s(k)} + h(S) \tag{20}$$

for every S and k if and only if

$$\frac{a(S,k)}{s(k)} = u(S) + v(k) \tag{21}$$

for some functions u and v.

Proof. If such $c(k)$ and $h(S)$ exist, substituting the decomposition of $d(S,k)$ gives

$$\frac{a(S,k)}{s(k)} = h(S) + \frac{c(k) - b(k)}{s(k)},$$

which has the form $u(S) + v(k)$. Conversely, if $a(S,k)/s(k) = u(S) + v(k)$, choose

$$c(k) = b(k) + s(k)v(k).$$

The required relation then holds with $h(S) = u(S)$. □

Because $h(S)$ is shared across candidates, it does not affect their ranking for a fixed audited model. For SCOUT, the candidate-only correction and scale are

$$c(k) = \frac{1}{K-1} \sum_{j \neq k} D(T_j, T_k), \qquad s(k) = \sigma(T_k).$$

Hence, it can remove candidate-wide proximity but cannot generally remove affinity specific to one audited-model–candidate pair. This limitation applies to candidate-only calibration generally rather than SCOUT alone. Shared model lineage may induce such an interaction, although the proposition does not imply that every lineage effect is nonseparable or unresolvable with additional information.

C.4 Limits of Output-Only History Recovery

Proposition 4 (Limits of output-only history recovery)**.** *Let h_1 and h_2 be two training histories, and let P_{h_1} and P_{h_2} denote the distributions they induce over the complete current response observation $\mathcal{Y}$ under a fixed prompting and decoding protocol. With equal prior probability on the histories, any output-only auditor distinguishing h_1 from h_2 has balanced accuracy at most*

$$\mathrm{BA} \le \frac{1 + \mathrm{TV}(P_{h_1}, P_{h_2})}{2}. \tag{22}$$

Proof. Let A be the set of observations for which the auditor selects h_1. Its balanced accuracy is

$$\begin{aligned} \mathrm{BA} &= \frac{1}{2} P_{h_1}(A) + \frac{1}{2}\big(1 - P_{h_2}(A)\big) \\ &= \frac{1}{2} + \frac{1}{2}\left(P_{h_1}(A) - P_{h_2}(A)\right) \\ &\le \frac{1 + \mathrm{TV}(P_{h_1}, P_{h_2})}{2}, \end{aligned}$$

by the definition of total variation distance. □

If the two histories induce identical current-output distributions, no output-only auditor can exceed chance-level balanced accuracy. The proposition is conditional. It bounds recoverability when two histories induce similar current-output distributions but does not assert that post-training necessarily makes those distributions similar. The source-switching results below show that the evaluated output-based readouts can lose the earlier signature. They do not estimate total variation or establish failure for every possible auditor. Consequently, abstention indicates insufficient evidence in current outputs rather than proof that a candidate was never used earlier in training.

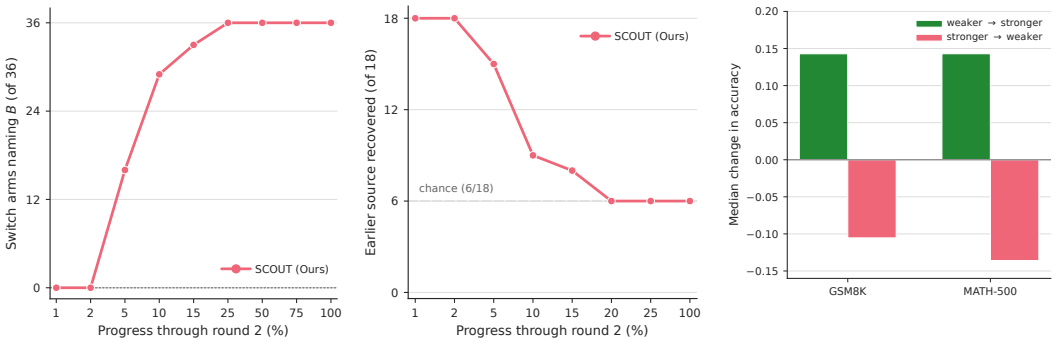
C.5 Source Switching Under Later SFT

Section 6 examines later training that does not introduce a new candidate source. In this section, we consider a second SFT round using responses from another candidate. Let A and B denote the first- and second-round sources, respectively.

Source Redirection We start from eighteen controlled students spanning three base models, three first-round sources, and two training sets. Continuing each student on either alternative source yields 36 switch arms, while continuing on A yields 18 same-source controls. Each second round uses 800 responses, and evaluation uses the probe set excluded from training.

Figure 11a shows rapid redirection toward B. SCOUT selects B in $16/36$ arms after 5% of the second round, $29/36$ after 10%, and $33/36$ after 15%. It selects B in every switch arm from 25% onward, while none of the same-source controls switches.

Earlier-Source Recovery To test whether A remains recoverable after the readout switches to B, we use a separate 18-arm replication crossing three base models with all six ordered source pairs. Each second round uses 1,000 responses. We remove B, recompute the representation and calibration statistics over the remaining three candidates, and test whether A ranks first. Random selection would recover A in $6/18$ arms. Figure 11b shows that SCOUT recovers A in $18/18$ arms at 1% and 2% of the second round, $15/18$ at 5%, and $6/18$ at the endpoint. Thus, after sufficient second-round SFT, SCOUT no longer recovers the earlier source reliably even after B is removed from the candidate pool.



(a) Attribution to the new source (b) Recovery of the earlier source (c) Capability change

Figure 11: **Source switching under a second SFT round. (a)** SCOUT attribution to B across 36 switch arms, with no switches among the 18 same-source controls. **(b)** Recovery of A across 18 arms after removing B. The dashed line marks chance. **(c)** Median endpoint accuracy change when moving to a stronger or weaker source.

Table 15: **Attribution to the second-round source during source switching.** Entries count the 18 switch arms attributed to B after each percentage of the second SFT round. Higher counts indicate stronger redirection, not better attribution. DistillDetect uses the pre-distillation base as its reference.

Method	1%	2%	5%	10%	15%	20%	25%	100%
Reference-based								
DistillDetect	0	0	2	11	12	14	14	14
Output-only								
PoS Templates	0	0	6	14	16	18	18	18
SCOUT (Ours)	0	0	7	15	17	17	17	18
− Scaling (C)	0	0	7	15	17	17	17	18
− Centering (D)	0	0	6	15	17	17	17	18

Capability Tradeoff For each benchmark, we define source strength using the median accuracy before the second round among students initially trained on that source. We call B stronger when its median is at least that of A, and weaker otherwise. Figure 11c shows that when B is stronger, median accuracy rises by 0.143 on both GSM8K and MATH-500. When B is weaker, it falls by 0.105 and 0.136, respectively.

The capability change depends on the relative strength of the two sources. Switching to a stronger B preserves or improves median benchmark accuracy, whereas switching to a weaker B reduces it. However, without a matched B-only baseline, these experiments do not isolate whether benefits acquired from A remain after switching. Section 6 instead considers subsequent training without source replacement and finds that the original signature remains detectable.

Redirection Across Readouts Table 15 compares readouts on the 18-arm replication. Every method selects B in a majority of arms by 10%. All output-only readouts select B in $18/18$ arms at the endpoint, showing that redirection is not specific to SCOUT or profile aggregation.

These results do not establish that the earlier history is erased or inaccessible to every possible auditor. They show that later SFT can redirect the signature expressed in current outputs, supporting the output-only limit formalized in Proposition 4.

D Additional Attribution Analyses

This appendix reports two additional analyses. First, it supplements the controlled evaluation with threshold-sweep and significance analyses. Second, it tests candidate-pool sensitivity using a separate panel of seven public R1-derived suspects.

Table 16: **Controlled detection ablation.** AUC is computed from student-level detection and false-positive rates. Confidence intervals are obtained by bootstrapping students, with all evaluation cells belonging to each sampled student retained together. At zero false positives on the corresponding aggregation axis, *Cells* reports the maximum number of correctly accepted source-present probe cells, whereas *Students* reports the maximum number of students correctly accepted on both probes. The two columns may use different thresholds. Higher is better.

Method	AUC [95% CI]	Cells	Students
Math, 1.5–4B students			
DistillDetect	0.8587 [0.7895, 0.9501]	26/38	13/19
PoS Templates	0.7230 [0.5346, 0.8726]	15/38	11/19
SCOUT (Ours)	1.0000 [1.0000, 1.0000]	37/38	19/19
− Scaling (C)	1.0000 [1.0000, 1.0000]	33/38	19/19
− Centering (D)	1.0000 [1.0000, 1.0000]	35/38	19/19
Conversation, 1.5–4B students			
DistillDetect	0.8255 [0.7729, 0.9197]	24/38	12/19
PoS Templates	0.8172 [0.6149, 0.9723]	27/38	12/19
SCOUT (Ours)	0.9695 [0.9169, 1.0000]	34/38	17/19
− Scaling (C)	0.9501 [0.8615, 0.9945]	25/38	14/19
− Centering (D)	0.9030 [0.7701, 0.9917]	25/38	14/19
Math, 7–8B students			
DistillDetect	0.9478 [0.8752, 1.0000]	23/42	11/21
PoS Templates	0.6485 [0.4671, 0.8277]	13/42	11/21
SCOUT (Ours)	0.9932 [0.9728, 1.0000]	38/42	20/21
− Scaling (C)	0.9478 [0.8821, 1.0000]	36/42	19/21
− Centering (D)	0.9410 [0.8413, 1.0000]	38/42	19/21

D.1 Controlled Detection Ablation

Table 16 separates score discrimination from the fold-fitted threshold transfer evaluated in Table 2. We sweep the rejection threshold after evaluating each student with its source present and removed. At every threshold, detection requires correct attribution on both probes, while a false positive requires acceptance after source removal under the same two-probe decision rule.

In the 1.5–4B math cohort, all three profile variants attain an AUC of 1.000 and detect all nineteen students at zero false positives, while SCOUT gives the highest cell-level count. In the conversation cohort, centering raises AUC from 0.903 to 0.950, and scaling raises it to 0.970 while increasing student detection from $14/19$ to $17/19$. In the 7–8B math cohort, SCOUT attains an AUC of 0.993 and detects $20/21$ students, exceeding the uncalibrated and centered profiles. These threshold sweeps diagnose score separation, whereas Table 2 remains the operational evaluation of threshold transfer.

Table 17 shows that individual source-removed probe cells can have $p^\star < 0.05$ even when the complete decision rule produces no false acceptance. For SCOUT, this occurs under teacher holdout in the 1.5–4B math cohort and under both holdouts in the 7–8B cohort. However, these significant cells do not align across both probes with the same selected candidate, so no source-removed student is accepted. In the conversation cohort, the minimum source-removed $p^\star$ for SCOUT remains above 0.05 under both holdouts. An upper endpoint of 1.00 indicates that at least one source-present probe cell has no significant top-two gap, preventing detection for the corresponding student.

D.2 Candidate-Pool Stress Test with Source Relatives

To test sensitivity to close alternatives, we construct a source-relative set containing DeepSeek-R1-Distill-Qwen-1.5B, 7B, 14B, and 32B and DeepSeek-R1-Distill-Llama-8B and 70B (Guo et al., 2025), together with s1.1-32B (Muennighoff et al., 2025b). When auditing one of these seven suspects, the expanded candidate pool combines the original ten candidates with the other six source-relative models, excluding the audited suspect. Each trial is therefore a sixteen-way attribution decision that includes DeepSeek-R1 and six alternative R1-derived models. This paired comparison

Table 17: **Adjusted significance values underlying controlled detection.** The source-present column reports the range of $p^\star = \min\{1, (K-1)p\}$, where smaller values provide stronger evidence for correct acceptance. The source-removed columns report the minimum under backbone and teacher holdout. A source-removed value below 0.05 identifies a significant gap in one probe cell but does not itself constitute a false acceptance, which also requires the fold-fitted threshold and $p^\star < 0.05$ on both probes with the same selected candidate. LLM judges are omitted because they do not produce score gaps or fitted thresholds.

Method	Source present $p^\star$ range	Source removed backbone-holdout minimum $p^\star$	Source removed teacher-holdout minimum $p^\star$
Math, 1.5–4B students			
DistillDetect	8.35×10^{-164}–1.00	4.18×10^{-60}	1.32×10^{-78}
PoS Templates	2.19×10^{-34}–1.00	8.22×10^{-29}	1.43×10^{-34}
SCOUT (Ours)	3.85×10^{-92}–1.00	0.402	1.02×10^{-6}
Conversation, 1.5–4B students			
DistillDetect	6.44×10^{-130}–1.00	1.11×10^{-82}	7.19×10^{-77}
PoS Templates	5.27×10^{-31}–1.00	0.708	3.80×10^{-16}
SCOUT (Ours)	3.53×10^{-124}–1.00	0.487	1.00
Math, 7–8B students			
DistillDetect	7.50×10^{-155}–1.00	1.11×10^{-62}	4.93×10^{-53}
PoS Templates	2.15×10^{-34}–1.00	4.68×10^{-33}	9.56×10^{-29}
SCOUT (Ours)	5.62×10^{-105}–1.00	7.58×10^{-16}	7.58×10^{-16}

Table 18: **Candidate-pool stress test with source relatives.** The original pool contains ten candidates. For each suspect, the expanded pool adds the other six models from the source-relative set described above, giving sixteen candidates. Each entry counts DeepSeek-R1 selections among seven suspects. DistillDetect uses the corresponding pre-distillation checkpoint. Higher is better.

Method	Original Pool	Expanded Pool
Reference-based		
DistillDetect, ref. $= S^{(0)}$	7/7	6/7
Output-only baselines		
PoS Templates	5/7	1/7
LLM judge (Mistral)	6/7	1/7
LLM judge (Solar)	7/7	1/7
SCOUT and ablations		
SCOUT (Ours)	7/7	1/7
− Scaling (C)	7/7	1/7
− Centering (D)	7/7	1/7

holds the suspect and audit prompts fixed while changing the candidate pool. Table 18 reports how often each method selects DeepSeek-R1. We use OMI because its audit prompts do not overlap the suspects' training data.

Every output-only method selects DeepSeek-R1 for only $1/7$ suspects with the expanded pool, including several methods that attain $7/7$ with the original pool. DistillDetect instead retains $6/7$ selections when given the corresponding pre-distillation reference. Thus, selecting DeepSeek-R1 from current outputs is highly sensitive to close source relatives, whereas access to a historical reference largely preserves source resolution. For SCOUT, this sensitivity is consistent with the candidate-only calibration limit in Proposition 3.

E Additional Trajectory Analyses

This appendix extends the trajectory analysis in Section 6 by documenting the released checkpoint sequences analyzed there and reporting the corresponding s1 results. All analyses use the same

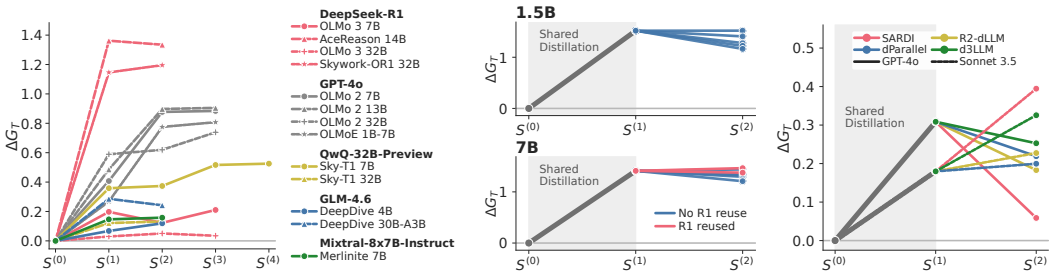
Table 19: **Training stages in the checkpoint trajectories.** $S^{(0)}$ is each lineage's pre-distillation checkpoint, and later columns name the training applied after the preceding checkpoint. Source headings indicate the designated teacher used for the reported margin rather than every contributor to the training mixture. Gray rows distinguish source and lineage anchors from their subsequent checkpoints. Dashes indicate that the trajectory ends.

Model or ladder	$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$	$S^{(4)}$
(a) Thirteen public ladders					
DeepSeek-R1					
OLMo 3 7B	Base	Dolci-Think SFT	DPO	RLVR	—
OLMo 3 32B	Base	Dolci-Think SFT	DPO	RLVR	—
Skywork-OR1 32B	Base	R1 distillation	RL	—	—
AceReason 14B	Base	R1 distillation	RL	—	—
GPT-4o					
OLMo 2 7B	Base	Tülu 3 SFT	DPO	RLVR	—
OLMoE 1B-7B	Base	Tülu 3.1 SFT	DPO	RLVR	—
OLMo 2 13B	Base	Tülu 3 SFT	DPO	RLVR	—
OLMo 2 32B	Base	Tülu 3 SFT	DPO	RLVR	—
QwQ-32B-Preview					
Sky-T1 7B	Base	QwQ SFT	PRIME RL	QwQ+self SFT	RLOO RL
Sky-T1 32B	Base	QwQ SFT	Preference optimization	—	—
GLM-4.6					
DeepDive 4B	Base	GLM-4.6 SFT	C-GRPO	—	—
DeepDive 30B-A3B	Base	GLM-4.6 SFT	C-GRPO	—	—
Mixtral-8x7B-Instruct					
Merlinite 7B	Base	Mixtral SFT	RLAIF	—	—
(b) Qwen2.5-Math lineages					
1.5B lineage					
R1-Distill-Qwen-1.5B	Base	R1 distillation	—	—	—
DeepScaleR-1.5B	Base	R1 distillation	GRPO	—	—
ZR1-1.5B	Base	R1 distillation	PRIME+RLOO	—	—
Nemotron-Research-1.5B	Base	R1 distillation	ProRL	—	—
DRPO-1.5B	Base	R1 distillation	DRPO	—	—
DisCO-1.5B-logL	Base	R1 distillation	DisCO	—	—
7B lineage					
R1-Distill-Qwen-7B	Base	R1 distillation	—	—	—
OREAL-7B	Base	R1 distillation	Outcome RL	—	—
Spiral-7B	Base	R1 distillation	Self-play RL	—	—
Skywork-OR1-7B	Base	R1 distillation	GRPO	—	—
DRPO-7B	Base	R1 distillation	DRPO	—	—
DisCO-7B-logL	Base	R1 distillation	DisCO	—	—
Light-R1-7B-DS	Base	R1 distillation	SFT on R1 outputs	—	—
R1-Distill-7B-ThinkPO	Base	R1 distillation	DPO with R1 chosen	—	—
(c) Dream-7B lineages					
Dream-v0-Instruct-7B	Base	Instruction tuning	—	—	—
SARDI	Base	Instruction tuning	SFT on GPT-4o-mini outputs	—	—
R2-dLLM	Base	Instruction tuning	Redundancy-aware SFT	—	—
dParallel	Base	Instruction tuning	Self-distillation	—	—
d3LLM	Base	Instruction tuning	Pseudo-trajectory distillation	—	—

thirteen-candidate pool, with candidate responses, profiles, and empirical nulls fixed across checkpoints and panels for each probe. Table 19 maps every checkpoint used in this analysis to its training stage. Panel **(a)** groups ladders by their designated source, panel **(b)** measures margins toward R1, and panel **(c)** reports separate GPT-4o and Sonnet 3.5 readings.

E.1 Thirteen Public Ladders

The thirteen ladders comprise checkpoint sequences from OLMo 2 (Walsh et al., 2025), OLMo 3 (OLMo Team, 2026), OLMoE (Muennighoff et al., 2025a), Skywork-OR1 (He et al., 2025), AceReason (Chen et al., 2025), Sky-T1 (NovaSky Team, 2025a;b), DeepDive (Zhang et al., 2026), and



(a) Thirteen public ladders (b) Qwen2.5-Math lineages (c) Dream-7B lineages

Figure 12: **Source-margin shifts from the pre-distillation state on s1.** Panels **(a)**, **(b)**, and **(c)** correspond to the OMI results in Figure 7 and show thirteen public ladders, two R1-distilled lineages and their descendants, and four post-training branches of Dream-7B, respectively. Positive values indicate a larger margin toward the indicated source than at $S^{(0)}$. Every post-distillation shift is positive, reproducing the OMI pattern. Shading marks distillation in **(b)** and **(c)**. Panel scales differ.

Merlinite (Sudalairaj et al., 2024). Their five designated sources are DeepSeek-R1 (Guo et al., 2025), GPT-4o (OpenAI, 2024a), QwQ-32B-Preview (Qwen Team, 2024), GLM-4.6 (GLM-4.5 Team et al., 2025), and Mixtral-8x7B-Instruct (Jiang et al., 2024). Figures 7a and 12a show the thirteen ladders on OMI and s1. Across thirteen ladders evaluated on two probes, $\Delta G_T > 0$ in all 68 post-distillation checkpoint–probe observations. These comprise 26 observations immediately after distillation and 42 observations from 21 later checkpoints.

E.2 R1 Descendant Lineages

We anchor the two R1 lineages of Section 5.2 at their Qwen2.5-Math bases (Yang et al., 2024a). Their distilled parents are DeepSeek-R1-Distill-Qwen-1.5B and 7B (Guo et al., 2025). The twelve descendants are DeepScaleR-1.5B (Luo et al., 2025), ZR1-1.5B (Zyphra, 2025), Nemotron-Research-1.5B (Liu et al., 2025b), DRPO-1.5B and 7B (Li et al., 2025a), DisCO-1.5B-logL and 7B-logL (Li et al., 2025b), OREAL-7B (Lyu et al., 2025), Spiral-7B (Liu et al., 2025a), Skywork-OR1-7B (He et al., 2025), Light-R1-7B-DS (Wen et al., 2025), and R1-Distill-7B-ThinkPO (Yang et al., 2025b). Figures 7b and 12b show positive ΔG_T for both distilled parents and all twelve descendants on both probes. Consequently, all 28 model–probe observations are positive. Seven of twelve descendants on OMI and three of twelve on s1 exceed their respective distilled parents, showing that the signature persists but does not consistently strengthen during subsequent training.

E.3 Masked Diffusion Lineage

The Dream-7B (Ye et al., 2025) analysis tests whether the contiguous PoS n-gram signal depends on left-to-right autoregressive generation. Dream-7B instead generates through iterative masked denoising. We evaluate its instruction-tuned checkpoint and four public post-training branches, SARDI (Jünger et al., 2026), R2-dLLM (Du et al., 2026), dParallel (Chen et al., 2026), and d3LLM (Qian et al., 2026), relative to the pre-distillation base.

Because its instruction data include responses from GPT-4o (OpenAI, 2024a) and Sonnet 3.5 (Anthropic, 2024), Figures 7c and 12c report separate margins toward both models. Both margins increase at $S^{(1)}$ after instruction tuning and remain positive at $S^{(2)}$ for all four branches on both probes. SARDI underwent additional SFT on GPT-4o-mini outputs between $S^{(1)}$ and $S^{(2)}$, which may affect its GPT-4o margin at $S^{(2)}$. Even when SARDI is excluded to account for this additional supervision, the GPT-4o margins at $S^{(2)}$ remain positive for all three other branches on both probes.

F Experimental Details

This appendix specifies the datasets, cohort construction, response generation, distillation, candidate pools, and scoring procedures underlying the retrospective, controlled, and trajectory analyses.

Table 20: **Controlled SFT configuration.**

Setting	Value
Objective	Completion-only causal language modeling
Fine-tuning method	Full-parameter fine-tuning
Epochs	3
Learning rate	1×10^{-5}
Schedule	Cosine with warmup ratio 0.05
Effective batch size	16 (batch size 4, accumulation 4)
Maximum sequence length	4,096 tokens
Training precision	bfloat16

Controlled cohorts The controlled evaluation in Section 5.3 compares a math cohort following DistillDetect (Rawat et al., 2026) with a conversation cohort constructed for this study to test whether attribution extends beyond math. Both begin as 24-student grids with the same backbone-source structure, using Gemma-3-4B-PT (Gemma Team et al., 2025), Llama-3.2-3B-Instruct (Grattafiori et al., 2024), Qwen-2.5-1.5B, and Qwen-2.5-3B (Yang et al., 2024b) as backbones. Within each grid, the source varies among GPT-OSS-120B (OpenAI, 2025a), Llama-3.3-70B-Instruct, and Qwen-3-8B (Yang et al., 2025a). The math grid trains each student on 1,000 responses from either OMI (Toshniwal et al., 2024) or s1 (Muennighoff et al., 2025b) and retains the nineteen students for which distillation improves performance over the corresponding base model as assessed on GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021). The conversation grid instead trains each student on 937 responses from either OASST1 (Köpf et al., 2023) or Dolly (Conover et al., 2023). Filtering Dolly and excluding its 200 audit prompts leaves 937 eligible prompts. The OASST1 training bank is also disjoint from its 200 audit prompts and is subsampled to the same size. This grid retains the nineteen students whose strict prompt-level IFEval accuracy (Zhou et al., 2023) on full responses exceeds that of the corresponding base model. An additional 24-student math grid uses Falcon3-7B-Base (Falcon-LLM Team, 2024), Llama-3.1-8B (Grattafiori et al., 2024), Mistral-7B-v0.3 (Jiang et al., 2023), and Qwen-2.5-7B (Yang et al., 2024b). Using the same sources and math datasets, it retains twenty-one students whose mean accuracy across GSM8K and MATH-500 exceeds that of the corresponding base model. The controlled candidate pool contains the three sources and Gemma-3-27B-it, an unexposed distractor that shares a model family with Gemma-3-4B-PT in the 1.5–4B grids. Attribution and significance use the probe dataset not used for SFT, whereas detection and abstention use both probe datasets.

Controlled SFT We train each controlled student using full-parameter SFT with a completion-only causal language-modeling objective. Prompt tokens are masked to -100, and each prompt–response pair forms one example. Examples are truncated rather than packed. Table 20 reports the shared configuration.

Prompt formatting Llama-3.2-3B-Instruct is the only controlled backbone that uses its tokenizer's chat template. All other controlled backbones use the plain-text format

`Problem:\n{question}\n\nSolution:\n`.

The same backbone-specific format is used for SFT and for generating controlled student responses.

Response generation For locally generated corpora, we generate one response per prompt with a model-prompt-specific seed, using each model's released generation configuration when available and the serving defaults otherwise. Consequently, temperature, top-p, top-k, repetition penalty, and end-of-sequence tokens may differ across models. We set the maximum output length to 4,096 tokens, except that the existing Qwen-3-8B (Yang et al., 2025a) training and candidate response banks use a 2,048-token limit. Every response that reaches either limit exceeds the first 2,000 characters analyzed by SCOUT, so the limit does not alter the analyzed prefix. For the utility screens used to select the controlled cohorts, we instead generate responses from the controlled students with a 16,384-token limit and score their full responses on GSM8K, MATH-500, and IFEval.

Candidate pools and response provenance The retrospective public-model evaluation in Section 5.2 uses domain-specific ten-candidate pools. For the math probes, the pool contains Gemma-

3-27B-it (Gemma Team et al., 2025), Llama-3.3-70B-Instruct, GPT-OSS-120B (OpenAI, 2025a), Claude-3.5-Sonnet (Anthropic, 2024), Claude Opus 4.5 and Claude Opus 4.6 (Anthropic, 2025c; 2026b), DeepSeek-R1 (Guo et al., 2025), o1 and o3 (OpenAI, 2024b; 2025b), and QwQ-32B-Preview (Qwen Team, 2024). Candidate responses for these probes come from the response bank released with DistillDetect (Rawat et al., 2026). For the conversation probes, we collect a new candidate response bank. Claude-3.5-Sonnet was no longer served through OpenRouter, so we replace it with its successor, Claude Sonnet 4 (Anthropic, 2025b). This retains a Sonnet-family candidate and the ten-candidate pool size while providing matched responses to the OASST1 and Dolly prompts. We generate all suspect and descendant responses in our pipeline. The math and conversation pools therefore have the same size but differ in one candidate and in response provenance.

Section 6 uses a thirteen-candidate pool for all trajectory panels. This pool extends the math ten-candidate pool with GPT-4o (OpenAI, 2024a), Mixtral-8x7B-Instruct (Jiang et al., 2024), and GLM-4.6 (GLM-4.5 Team et al., 2025) so that all documented sources in the panels are included without changing the pool between trajectories. The original ten candidate responses come from the DistillDetect response bank, and we generate responses for the three added candidates. For each probe, the candidate responses, profiles, and empirical nulls remain fixed across all checkpoints and panels.

Public models and checkpoints The retrospective panel in Section 5.2 contains the Qwen2.5-Math-1.5B and Qwen2.5-Math-7B bases (Yang et al., 2024a), their DeepSeek-R1-Distill-Qwen parents (Guo et al., 2025), and twelve descendants. The 1.5B descendants are DeepScaleR-1.5B (Luo et al., 2025), ZR1-1.5B (Zyphra, 2025), Nemotron-Research-1.5B (Liu et al., 2025b), DRPO-1.5B (Li et al., 2025a), and DisCO-1.5B-logL (Li et al., 2025b). The 7B descendants are OREAL-7B (Lyu et al., 2025), Spiral-7B (Liu et al., 2025a), Skywork-OR1-7B (He et al., 2025), DRPO-7B (Li et al., 2025a), DisCO-7B-logL (Li et al., 2025b), Light-R1-7B-DS (Wen et al., 2025), and R1-Distill-7B-ThinkPO (Yang et al., 2025b). Light-R1-7B-DS received further SFT on R1 outputs, and R1-Distill-7B-ThinkPO used R1 outputs as chosen responses in DPO. The other ten underwent reward-based or self-play post-training without additional R1 responses.

The trajectory analysis in Section 6 uses thirteen lineages composed entirely of publicly released checkpoints. These include OLMo 2 (Walsh et al., 2025), OLMo 3 (OLMo Team, 2026), OLMoE (Muennighoff et al., 2025a), Skywork-OR1 (He et al., 2025), AceReason (Chen et al., 2025), Sky-T1 7B and 32B (NovaSky Team, 2025a;b), DeepDive (Zhang et al., 2026), and Merlinite (Sudalairaj et al., 2024). The publicly released Dream-7B lineage comprises the base and instruction-tuned successor (Ye et al., 2025) and four post-training branches, SARDI (Jünger et al., 2026), R2-dLLM (Du et al., 2026), dParallel (Chen et al., 2026), and d3LLM (Qian et al., 2026). We generate their audit responses but do not train or modify the checkpoints. Appendix E lists each checkpoint, designated source, and training stage. Finally, the public-suspect evaluation in Appendix D includes DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Qwen-14B, DeepSeek-R1-Distill-Qwen-32B, DeepSeek-R1-Distill-Llama-8B, DeepSeek-R1-Distill-Llama-70B (Guo et al., 2025), and s1.1-32B (Muennighoff et al., 2025b). All seven are evaluated on OMI. For s1, we exclude s1.1-32B because its fine-tuning set, s1K-1.1, contains every s1 probe prompt paired with a DeepSeek-R1 response.

Representation and scoring We retain the first 2,000 characters of each response and process them with NLTK 3.9.1 (Bird & Loper, 2004), using `word_tokenize` followed by `pos_tag`. We then count contiguous PoS n-grams for $n \in \{3, 4, 5\}$. We use no lowercasing or IDF weighting. For each probe and candidate pool, the feature vocabulary is fitted once on all candidate responses. The ℓ_2 norm used in cosine distance includes all PoS n-grams in the evaluated response, including those absent from the candidate vocabulary. Counts and distances are computed in double precision. Appendix B examines the response window and related representation choices.

Evaluation For all score-based experiments, we use the evaluator released with DistillDetect (Rawat et al., 2026) to aggregate prompt-level scores. For controlled detection and abstention in Section 5.3, we also use its threshold-fitting and significance-testing procedures. Thresholds are fitted on the non-held-out cells of each evaluation fold. Student-backbone holdout excludes all students based on the audited backbone, while teacher holdout excludes every student associated with the audited teacher. For detection, the evaluator identifies the two leading candidates and uses their per-prompt score differences to test whether the mean gap exceeds the fold-fitted threshold. It then

applies a one-sided one-sample t-test or Wilcoxon signed-rank test according to a Shapiro–Wilk normality check and computes $p^\star = \min\{1, (K-1)p\}$. A detection is reported only when $p^\star < 0.05$ on both probes and both probes select the same candidate.

LLM Judge Protocol We query Mistral Medium 3.1 (Mistral AI, 2025) and Solar Pro 4 (Upstage, 2026) through OpenRouter with temperature 0 and a maximum of 64 output tokens. Each call contains the audit prompt, one response from every candidate, and one suspect response. Candidate and suspect responses are truncated to their first 2,000 characters. Candidate identities are replaced by letter labels, and the mapping between labels and candidates is deterministically reshuffled for every suspect and prompt. We map each parsed label back to its candidate identity before aggregation and do not retry outputs that cannot be parsed. Unparseable outputs count as `NONE` in open-set trials and are excluded in closed-set trials. The most frequent result across 200 prompts becomes the model-level verdict for that probe. Ties yield no verdict.

LLM Judge Prompts Both judges receive the shared prompt below followed by one of two instructions. The retrospective and public-suspect evaluations use the closed-set instruction because they require a candidate choice. Controlled detection uses the open-set instruction because teacher-removed trials require abstention.

Shared Prompt

```
You are comparing writing styles.

Below are responses written by several CANDIDATE models to the
    same prompt, followed by a
response written by a SUSPECT model to that prompt.

Judge only from writing style, structure, and phrasing. Ignore
    correctness, factual content,
the specific reasoning path, and whether two responses reach the
    same conclusion.

PROMPT:
{question}

CANDIDATE RESPONSES:
{candidates}

SUSPECT RESPONSE:
{suspect}
```

Closed-Set Instruction

```
Which candidate response is stylistically most similar to the
    suspect response?
You must choose exactly one candidate.

Reply with JSON only:
{"choice": "<candidate label>"}
```

Open-Set Instruction

```
Identify the candidate response that is stylistically most similar
    to the suspect response.
If none of the candidate responses is sufficiently similar to
    support a plausible stylistic match,
```

```
answer NONE.

Reply with JSON only:
{"choice": "<candidate label>"}

or

{"choice": "NONE"}
```

Source switching Appendix C reports two independent source-switching campaigns. The 36-arm campaign starts from controlled students trained with a maximum sequence length of 4,096 and uses 800 second-round responses per arm. The 18-arm campaign starts from earlier first-round checkpoints whose effective maximum sequence length was 1,024 and uses 1,000 second-round responses per arm. Both campaigns cover Gemma-3-4B-PT, Llama-3.2-3B-Instruct, and Qwen-2.5-3B with the same three controlled sources. We report them separately because their first-round checkpoints and second-round training sets differ.