ANCESTRAL MEMBERSHIP AUDITING: TRACING SOURCE USE THROUGH SYNTHETIC DESCENDANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic training data can replace original records while retaining information about the sources that generated them. We study *ancestral membership auditing*: determining whether synthetic descendants of a known source were used for downstream fine-tuning when the source itself is excluded from training and the actual descendants remain hidden. We identify the *Sibling Transfer Effect* (STE), where independently regenerated examples from member sources exhibit more favorable average changes in answer likelihood than those from nonmember sources. We audit this effect using source-grounded question-answer probes and source-level likelihood readouts with base or reference calibration. Across eight model families on Enron and BillSum, five-reference calibration achieves mean AUCs of 0.9240 and 0.9477 with matched generators and 0.8328 and 0.8504 under generator mismatch. Removing exact matches to training records preserves detectability, and the strongest signal occurs among non-exact siblings sharing source evidence. STE can also appear through relative retention, where member probes deteriorate less even when their absolute likelihood decreases. These results show that synthetic transformation can change training records without necessarily erasing an auditable footprint of source participation.

1 INTRODUCTION

Synthetic instruction generation turns language-model outputs into training data (Wang et al., 2023). Conditioning generation on source text further allows documents to be transformed into instruction-response examples for model adaptation (Nayak et al., 2024; Cheng et al., 2024). An email or bill can therefore supply training content without appearing in the fine-tuning inputs. Auditing whether the original document was itself a training record misses this form of participation. We ask: *can a source's use be detected when only its synthetic descendants enter downstream fine-tuning?*

We call this problem *ancestral membership auditing*. We use *provenance depth* n to count the transformations separating a source from the records used for downstream training (Figure 1). Direct record membership corresponds to $n = 0$. We study $n = 1$, where a source conditions synthetic generation, and only the resulting descendants are used for fine-tuning. This setting also differs from membership auditing over related samples. User-level inference can infer participation from samples associated with a user (Kandpal et al., 2024), whereas here the related training samples are

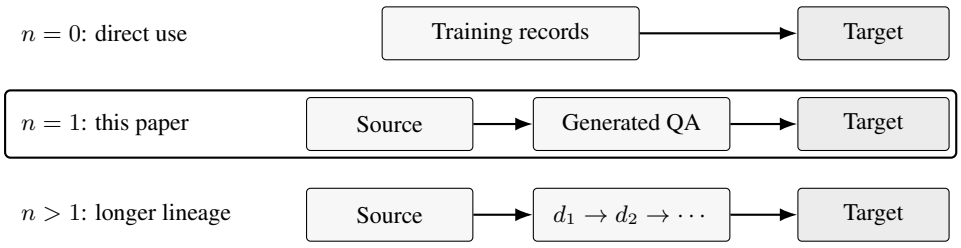


Figure 1: Source use at different provenance depths. We evaluate $n = 1$, where the source conditions synthetic generation but is absent from target fine-tuning. Longer lineages are not evaluated.

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.

not available to the auditor. They are hidden descendants produced through a source-conditioned generation process. Longer lineages with $n > 1$ are outside our empirical scope.

The main difficulty is that the records linking a source to the target model are hidden. The auditor does not know which source content the provider selected or how that content was expressed in training examples. Prior work shows that membership information can survive distillation and synthetic-data pipelines (Jagielski et al., 2023; Meeus et al., 2025; Akkus et al., 2025), but our question concerns the provenance of one candidate source rather than whether a downstream model used synthetic data in general (Wu et al., 2025). An ancestral membership audit must therefore detect source participation without observing or recovering the descendants used for training.

We find that training on hidden descendants leaves a measurable footprint on other examples generated independently from the same source. We call this phenomenon the *Sibling Transfer Effect* (STE). Regenerated siblings from member sources exhibit more favorable average changes in answer likelihood than siblings from nonmember sources, even though the regenerated siblings were not used to train the target. The effect can appear as an increase in likelihood or as *relative retention*, where member probes deteriorate less than nonmember probes. STE provides a measurable link between hidden training descendants and independently regenerated examples that share their source. We use regenerated siblings to audit this signal at the source level. Across Enron and BillSum, all eight target-model families exhibit detectable source participation under both matched and mismatched generators. Removing audit examples that exactly match training descendants preserves detectability, while randomly reassigning sibling families across sources reduces the signal to chance. The signal is also stronger for non-exact siblings grounded in evidence shared with the hidden training descendants. These results suggest that STE reflects source-grounded transfer rather than recurrence of the exact training records.Our contributions are

- We formulate membership of a source whose synthetic descendants, rather than the source itself, are used for downstream fine-tuning. We study provenance depth $n = 1$, where the training descendants remain hidden from the auditor.
- We identify a membership signal on independently regenerated siblings. Exact-match removal, genealogy shuffling, and evidence-level controls show that the signal does not require recurrence of the target model's training records and is strongest when siblings share source evidence with the hidden descendants.
- We develop a sibling-based source audit and evaluate the signal across eight model families, two domains, and matched and mismatched generators. We further characterize its behavior under generator mismatch, evidence overlap, relative retention, and different auditing resources.

2 Related Work

Data provenance and tracing. Data provenance research studies dataset origins, attribution, licensing, and reuse (Longpre et al., 2024; Hohensinner et al., 2026). These approaches primarily document or trace relationships among data artifacts. We study a behavioral provenance question: whether a downstream model reveals that a known source contributed to training through synthetic descendants, even when the source itself was not used as a training record.

Membership beyond individual records. Behavioral provenance auditing is closely related to membership inference. Membership inference traditionally asks whether a candidate record was used for training (Shokri et al., 2017), and has been extended to documents (Meeus et al., 2024), datasets (Maini et al., 2024), and users (Kandpal et al., 2024). User-level inference shows that participation can be inferred from related samples not used for training. Our setting differs in how these related samples arise and what the auditor observes. The descendants used for training are hidden and produced through a source-conditioned generation process, while the auditor observes the source and constructs a separate probe family. Ancestral membership therefore asks whether these latent descendants connect a known source to downstream training.

Information flow through synthetic data. Synthetic-data introduce the lineage that separates the audited source from the downstream training records. Prior work shows that such pipelines do not necessarily remove membership information. Membership has been traced from a student's behavior to the teacher's training set through distillation (Jagielski et al., 2023), and from released synthetic text to records used to train its generator (Meeus et al., 2025). Other work studies privacy exposure

when generated data are used for downstream fine-tuning (Akkus et al., 2025), while synthetic artifact auditing asks whether a downstream artifact used synthetic data (Wu et al., 2025). These settings establish information flow across synthetic-data pipelines but target different provenance relations. We ask whether synthetic descendants of one candidate source were used to train the downstream model. The auditor knows the source but not the descendants connecting it to the target.

Membership readouts and evaluation. The preceding work concerns what participation is audited; a separate line of research studies how membership signals are measured. Membership attacks use likelihood ratios and reference models (Carlini et al., 2022; Zarifzadeh et al., 2024), token-level statistics (Shi et al., 2024; Zhang et al., 2025), and comparisons with generated or neighbouring examples (Mattern et al., 2023). Other readouts include WBC (Chen et al., 2026), Error Zone scoring (Ilić et al., 2026), and in-context probing (Lu et al., 2026). We use these methods as complementary readouts of source participation rather than treating the likelihood statistic itself as the auditing target. Prior studies also identify distribution shift, synthetic-text artifacts, and training variation as confounders in membership evaluation (Duan et al., 2024; Maini et al., 2024; Naseh & Mireshghallah, 2025; Hayes et al., 2025). Our evaluation therefore separates matched and mismatched generators and compares methods on shared source splits.

3 Ancestral Membership Auditing

3.1 Task and threat model

A provider generator G_P produces synthetic descendants from a source s. Let $Z_{\text{train}}^{(G_P)}(s)$ denote the descendants selected for fine-tuning, and let $\mathcal{M}_T$ contain the sources whose descendants train target model T. Starting from a pretrained base B, the target is fine-tuned on

$$\mathcal{D}_T = \biguplus_{s \in \mathcal{M}_T} Z_{\text{train}}^{(G_P)}(s), \qquad m_T(s) = \mathbf{1}\{s \in \mathcal{M}_T\}. \tag{1}$$

Ancestral membership auditing infers $m_T(s)$. By construction, $s \notin \mathcal{D}_T$; membership is induced only through synthetic descendants generated from s. We instantiate these descendants as source-grounded QA pairs $z = (q, a)$. The source text itself is excluded from downstream fine-tuning, including as additional context. The membership label therefore records source provenance rather than direct inclusion of the source or recurrence of an identical QA record.

The auditor knows the candidate source s and can query conditional likelihoods for supplied answers, but cannot access target weights, gradients, or the descendants used for training. Calibrated variants also use a pretrained base, while reference calibration additionally requires auxiliary data and fine-tuning resources. Figure 2 illustrates the provider and auditor paths.

We consider two generator settings. With a *known generator*, the auditor reuses G_P through fresh generation calls. With an *unknown generator*, the auditor selects its own generator G_A without knowing G_P. Same-generator evaluation uses $G_A = G_P$, while cross-generator evaluation uses $G_A \neq G_P$. In both settings, the provider's actual training descendants remain hidden.

3.2 Reading participation through siblings

Conditioned on s, the auditor uses G_A to construct a source-grounded probe family $Z_{\text{audit}}(s) = (z_1, \ldots, z_L)$. These probes are *siblings* of the hidden training descendants because both are generated from the same source, although the auditor does not observe or reproduce the provider's generation process. We use these siblings to measure the *Sibling Transfer Effect* (STE). STE refers to the membership-dependent difference in sibling behavior rather than to a particular scoring function: siblings from member sources exhibit more favorable average changes in answer likelihood than those from nonmember sources. To read this effect while accounting for probe difficulty, we score the same answers under the target and calibration models and aggregate the resulting likelihoods by source. For a QA pair with answer tokens $a = (a_1, \ldots, a_n)$, define

$$\ell_M(q, a) = -\frac{1}{n} \sum_{t=1}^{n} \log p_M(a_t \mid \text{fmt}(q), a_{<t}), \qquad \bar{\ell}_M(s) = \frac{1}{L} \sum_{j=1}^{L} \ell_M(z_j). \tag{2}$$

Here $\text{fmt}(q)$ is the fixed question prefix. Only answer content tokens are scored, excluding EOS. We normalize each QA by its answer length and then average the QA scores equally within each source.

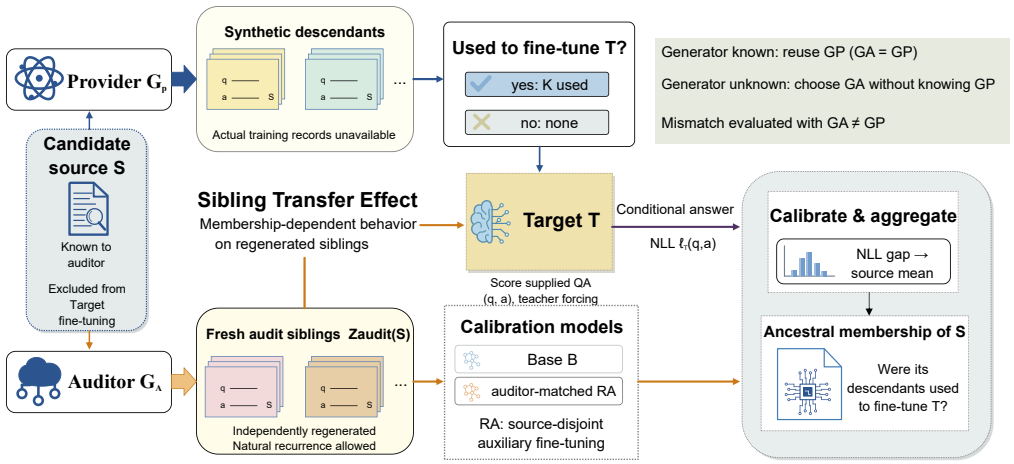


Figure 2: Auditing through regenerated siblings. Hidden descendants train the target without the raw source. The auditor independently regenerates QA siblings from the candidate source and compares their answer likelihoods under the target and calibration models. Reference models follow the auditor's generator. STE denotes the membership-dependent behavior observed on regenerated siblings.

Two basic readouts are

$$\mathrm{S}_{\mathrm{Target}}(s) = -\bar{\ell}_T(s), \tag{3}$$

$$\mathrm{S}_{\mathrm{MSG}}(s) = \bar{\ell}_B(s) - \bar{\ell}_T(s). \tag{4}$$

Target-only measures predictability under the fine-tuned model, while MSG measures adaptation on the same siblings relative to the pretrained base. Higher scores indicate more member-like behavior.

With the same normalization, the protocol-matched UserInf score (Kandpal et al., 2024) equals MSG (Appendix B). The distinction therefore lies in the auditing setting and probe construction rather than in this score: ancestral membership starts from a known source whose training descendants are hidden, and the auditor independently constructs siblings from that source.

Separate generation calls can reproduce QA records by chance. Our generation procedure prioritizes diversity before assigning training and audit roles while retaining repetitions when necessary (Appendix A.2). We use training-record filtering only for the oracle diagnostics in Section 4.4; practical auditing does not require knowledge of overlap with the hidden training descendants.

3.3 Reference calibration for the auditor's generator

Base calibration compares the target with an unadapted model. Reference calibration instead compares T with models adapted to the auditor's synthetic QA distribution. We fine-tune $R_{A,1}, \ldots, R_{A,r}$ from the same base using G_A auxiliary source cohorts that are disjoint from the evaluation sources. The resulting score is that

$$\mathrm{S}_{\mathrm{Multi}r}(s) = \frac{1}{r}\sum_{i=1}^{r} \bar{\ell}_{R_{A,i}}(s) - \bar{\ell}_T(s). \tag{5}$$

The reference models account for behavior associated with the auditor's generation process without requiring access to the provider's hidden descendants. We use $r = 5$ in the primary Multi5 setting. Reference-count comparisons use fixed prefixes $(R_{A,1}, \ldots, R_{A,r})$, beginning with $R_{A,1}$ for $r = 1$. The ensemble averages NLLs, and higher scores indicate stronger evidence of ancestral membership.

4 Experiments and Analysis

4.1 Setup

Data and models. Each domain contains 1,000 sources, Enron emails, or BillSum bill texts, split into 500 members and 500 nonmembers. Targets train on four QA descendants per member (2,000

Table 1: Source-level auditing: means over eight families and three same-generator or six cross-generator pairs. TPR_5/TPR_1 use 5%/1% FPR. Values are fractions; bold marks the highest point estimate per dataset, alignment, and metric.

	Same generator			Cross generator		
Method	AUC	TPR_5	TPR_1	AUC	TPR_5	TPR_1
Enron						
Target-only	0.7877	0.3067	0.0858	0.7076	0.1939	0.0655
Min-K% (20%)	0.8086	0.3200	0.0716	0.7162	0.2007	0.0546
Min-K%++ (20%)	0.6651	0.2322	0.0865	0.5987	0.1489	0.0548
UserInf	0.8594	0.5331	0.1240	0.7680	0.3554	0.1319
WBC	0.8683	0.3920	0.1168	0.7957	0.2789	0.0868
EZ-MIA	0.6844	0.1765	0.0435	0.6639	0.1429	0.0360
RMIA	0.8911	0.5862	**0.3684**	0.7991	0.4065	**0.2216**
ICP-MIA	0.7786	0.3312	0.1035	0.6909	0.2096	0.0788
MSG	0.8739	0.4485	0.1017	0.7921	0.3037	0.0843
Multi5	**0.9240**	**0.6863**	0.3029	**0.8328**	**0.4892**	0.2096
BillSum						
Target-only	0.8088	0.3344	0.1469	0.7027	0.1805	0.0622
Min-K% (20%)	0.8158	0.3479	0.1388	0.7002	0.1640	0.0539
Min-K%++ (20%)	0.6926	0.2526	0.0882	0.6034	0.1406	0.0429
UserInf	0.8877	0.6088	0.3979	0.7835	0.3610	0.1971
WBC	0.8733	0.3974	0.1107	0.7929	0.2472	0.0691
EZ-MIA	0.7573	0.2612	0.0699	0.7173	0.1642	0.0399
RMIA	0.8991	0.6121	0.3449	0.8005	0.3734	0.1802
ICP-MIA	0.8015	0.3945	0.2090	0.6911	0.2243	0.0798
MSG	0.8841	0.4640	0.1609	0.7861	0.2597	0.0725
Multi5	**0.9477**	**0.8117**	**0.5031**	**0.8504**	**0.5587**	**0.3003**

records); auditing uses 16 siblings per source. Each of five references trains on four descendants from its own 500-source cohort. Reference cohorts are mutually disjoint and exclude evaluation sources. Source assignments and QA banks are shared across model families within each domain. Three generators, Qwen, Granite, and Phi, serve as both provider and auditor. We evaluate all nine ordered generator pairs with eight model families: DeepSeek-Coder, Qwen3, SmolLM2, OLMo2, RedPajama, OpenLLaMA, Falcon, and Pythia. Targets and references use LoRA fine-tuning for five epochs with a fixed final checkpoint. More settings are specified in Appendix A.

Comparisons and evaluation. Table 1 compares ten readouts using the same audit probes, including token-based methods, sequence UserInf (Kandpal et al., 2024), WBC (Chen et al., 2026), EZ-MIA (Ilić et al., 2026), offline RMIA (Zarifzadeh et al., 2024), ICP-MIA (Lu et al., 2026), and our scores calibrated with the base or references. Min-K% and Min-K%++ use a fixed 20% token fraction (Shi et al., 2024; Zhang et al., 2025). Source-level adaptations and requirements for auxiliary resources are given in Appendix B.

We compute source-level AUC and TPR at 5%/1% FPR per configuration, then average equally across eight families and either three same-generator or six cross-generator pairs. Each family's same-generator ROC averages its three matched-generator ROCs at common FPRs. Paired bootstrap comparisons resample sources, retain their sibling families, and share resamples across methods and configurations. Intervals are conditional on the trained models (Appendix C).

4.2 Source participation is detectable across families and domains

Target-only likelihoods separate members from nonmembers across both domains and generator conditions (Table 1); base calibration strengthens the readout. UserInf and other adapted attacks also recover participation, so the signal is not specific to Multi5. Figure 3 resolves same-generator ROC comparisons by target family.

Multi5 has the highest mean AUC in all four dataset and alignment settings. Its paired gains over MSG on Enron are 0.0502 (95% CI $[0.0375, 0.0629]$) with the same generator and 0.0407 ($[0.0220, 0.0589]$) under mismatch. RMIA has higher TPR at 1% FPR on Enron, illustrating an operating-point tradeoff. Per-family comparisons appear in Appendices C.2 and H.

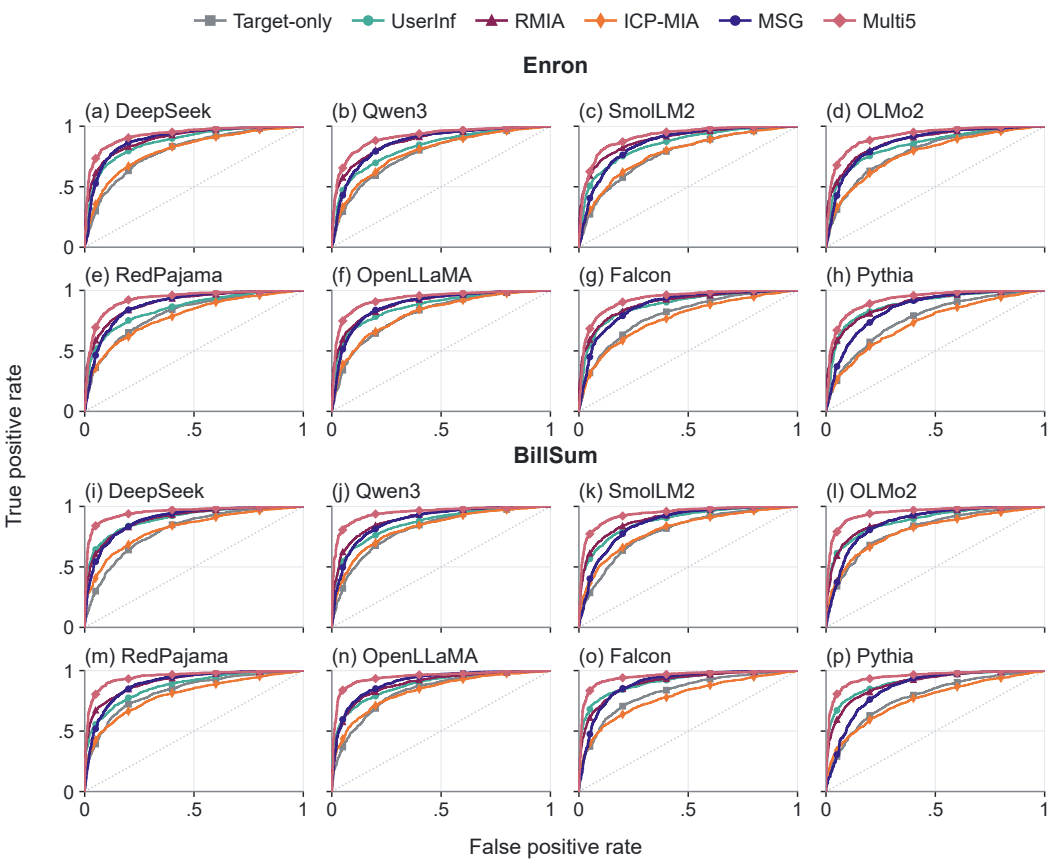


Figure 3: Same-generator ROC comparisons on Enron (top two rows) and BillSum (bottom two). Each curve averages three matched-generator ROCs at common FPRs. Six readouts are shown; all ten appear in Table 1 and Appendix C.5. Cross-generator results are in Table 1 and Figure 4.

4.3 Generator mismatch is directional

The mismatch penalty depends on the direction of transfer (Figure 4). Qwen-generated training descendants audited with Phi siblings produce stronger Multi5 readouts than the reverse pairing in both domains. All six off-diagonal means remain above chance. Matching the synthesis process improves detectability, while shared source content supports auditing across different generators. Reference training follows the auditor in every comparison.

4.4 Siblings reveal more than the recurrence of identical records

Why not score the source directly? Sibling MSG exceeds Direct-S in both domains and generator conditions for every family (Table 2, A). Same-generator gains have positive paired confidence intervals in both domains (Appendix E). Both readouts use the same base and target, but differ in probe format, token coverage, and query budget.

Does the signal require repeated training QA? A researcher oracle removes audit QA identical to actual training records. Multi5 retains nearly its natural-regeneration AUC in all four settings (Table 2, B), whereas Base-only and shuffled source–sibling correspondence remain near chance. Detectability thus persists beyond identical records; Section 4.8 examines the evidence carried by these probes.

4.5 Probe aggregation, calibration resources, and training progress

Figure 5 varies probe count, reference count, target epoch, and training members. Across eight families, increasing L using nested subsets of the same 16-probe bank improves mean AUC in both

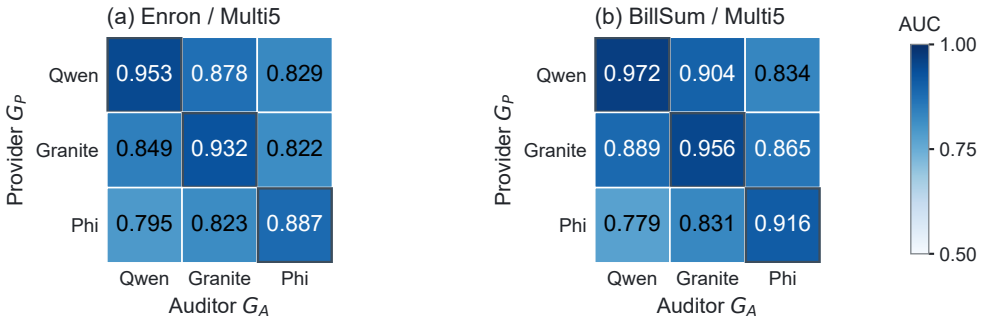


Figure 4: Generator mismatch under Multi5. Rows are providers G_P and columns auditors G_A; each entry averages AUC over eight target families. Outlined diagonal cells are same-generator conditions. The two domains share one color scale, and references follow G_A.

Table 2: Critical controls (mean AUC). A: Direct-S versus sibling MSG. Direct-S averages three provider targets, has no auditor axis, and differs in input and token/query budgets. B: Multi5 with natural probes, exact matches removed by an oracle, or shuffled genealogy.

A. Direct source versus siblings

Dataset	Direct-S	Siblings, same G	Siblings, cross G
Enron	0.7312	0.8739	0.7921
BillSum	0.7372	0.8841	0.7861

B. Recurrence and correspondence

	Enron		BillSum	
Condition	Same G	Cross G	Same G	Cross G
Natural siblings	0.9240	0.8328	0.9477	0.8504
Exact matches removed (oracle)	0.9207	0.8310	0.9468	0.8499
Shuffled genealogy	0.4994	0.4995	0.4996	0.4997
Base-only	0.5161	0.5161	0.4976	0.4976

domains. Adding references along the fixed prefix $R_{A,1:r}$ also helps, with diminishing increments near five references.

Epoch trajectories show the participation signal strengthening during fine-tuning. The same final references calibrate every target checkpoint: their advantage emerges later, while MSG can plateau or decline slightly between the final two epochs under mismatch.

Number of training members. For DeepSeek–Enron, six scales vary m from 5 to 500 with $K = 4$, $L = 16$, and 500 fixed nonmembers. Multi5's same-generator AUC is 0.8291 versus MSG's 0.9538 at $m = 25$, but 0.9306 versus 0.8972 at $m = 500$. Under both generator conditions, Multi5 is lower at tested $m \leq 50$ and higher at $m \geq 100$. References still train on 500 sources, and optimizer steps vary with m, so this does not isolate a causal scale effect. Results at $m = 5$ are exploratory (Appendix G.4).

4.6 The readout transfers to alternative pretrained bases

Across all target–base pairings, substituted bases retain a participation signal with mean AUC close to matched bases under the common BPB readout (Table 3). Full matrices and the separate Other7 ensemble appear in Appendix D.

4.7 Understanding the Sibling Transfer Effect

4.7.1 Relative retention can reveal participation

Let $g_T(s) = \mathsf{S}_{\mathrm{MSG}}(s)$ be adaptation gain relative to the pretrained base, with member/nonmember means $\overline{g}_M/\overline{g}_N$. Positive $\overline{g}_M$ indicates average absolute improvement. When $\overline{g}_N < \overline{g}_M < 0$, both

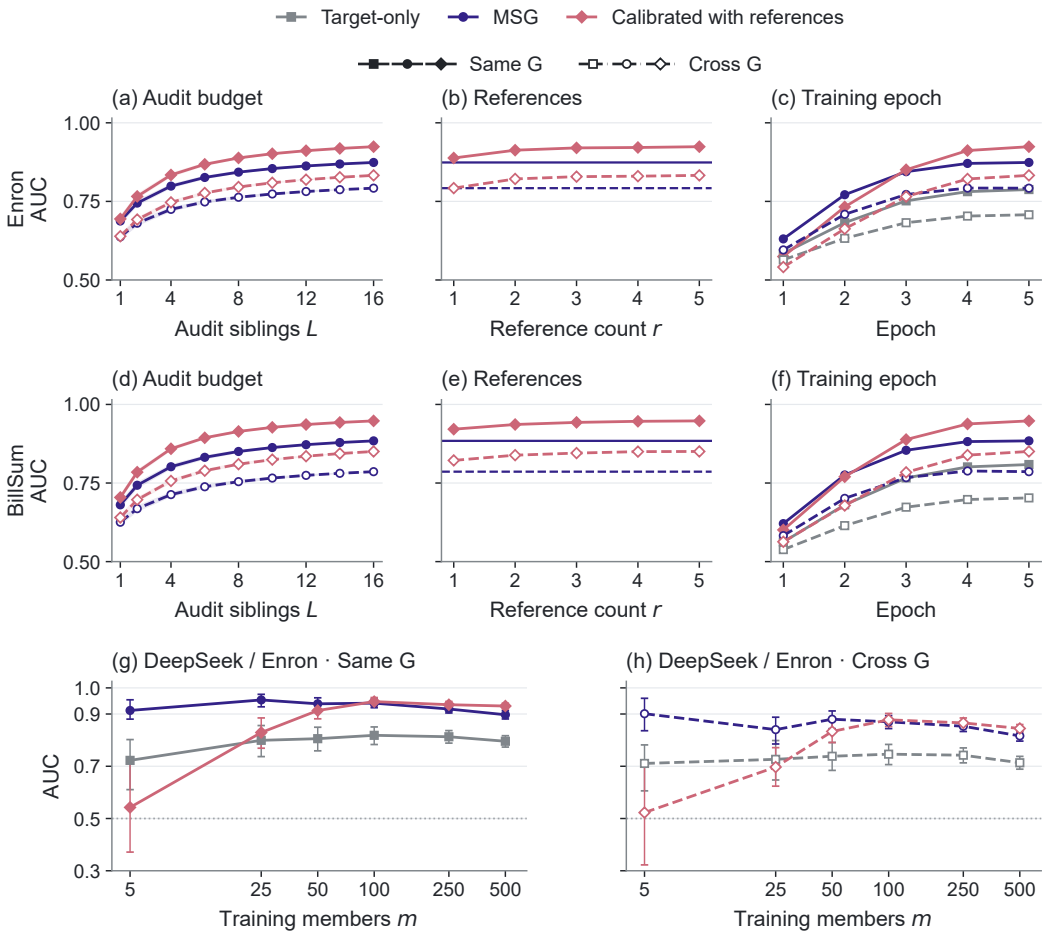


Figure 5: Joint sensitivity analysis. (a–f) Eight-family means on Enron/BillSum as L, r, and epoch vary. Reference counts follow R1→Multi2→ $\cdots$ →Multi5, with horizontal MSG baselines. Budget/epoch panels use Multi5; epoch panels keep final references. Budget shading: P05–P95 over QA subsets. (g,h) DeepSeek–Enron, varying training members m with 500 fixed nonmembers and reference cohorts of 500 sources; bars: pointwise 95% CIs from resampling sources. The m axis is logarithmic. Dotted: chance; solid/filled versus dashed/open: same/cross G.

Table 3: Substitution of the auxiliary base. Matched and substituted AUCs use the same BPB calibration, averaging the eight diagonal and 56 off-diagonal entries, respectively. Full 8×8 matrices are in Appendix D.

Dataset	Generator	Matched base	Substituted base
Enron	Same G	0.8584	0.8495
Enron	Cross G	0.7823	0.7719
BillSum	Same G	0.8528	0.8435
BillSum	Cross G	0.7596	0.7486

groups deteriorate but member siblings retain more likelihood. This *relative retention* is a favorable difference associated with membership despite absolute degradation.

Figure 6 (top) plots both group means for each configuration. Relative retention occurs in 3 of 24 same-generator configurations per domain, versus 28 of 48 cross-generator configurations on Enron and 32 of 48 on BillSum. An auditable signal can therefore persist despite decreasing absolute likelihood, especially under mismatch (Appendix F).

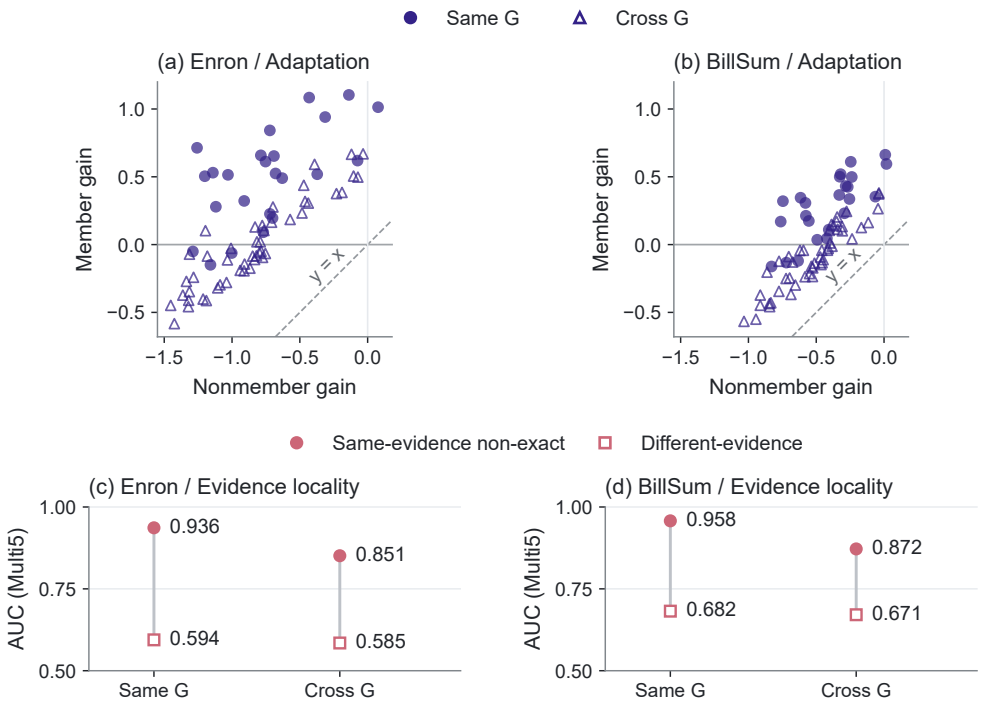


Figure 6: STE form and locality. Top: nonmember (x) and member (y) mean gains for 72 configurations per domain; $y = x$ marks equal gains and $y = 0$ zero member gain. Filled circles/open triangles denote same/cross G. Bottom: Multi5 AUC on common eligible sources, comparing siblings sharing evidence but differing from provider QA with siblings grounded in different evidence. Equal QA budgets are tested in Appendix F.

4.8 Transfer is strongest within shared evidence

Using the recorded evidence annotations, we compare siblings that share evidence with provider examples but do not exactly match them against siblings associated with different evidence units. Both strata use the same eligible sources. Figure 6 (bottom) shows a stronger signal for shared evidence under both generator conditions. Under cross-generator auditing, the respective AUCs are 0.8511 versus 0.5846 on Enron and 0.8719 versus 0.6710 on BillSum. The ordering persists in a separate control that equalizes QA budgets (Appendix F).

The audit unit is the source, but the strongest signal follows evidence shared by training descendants and regenerated siblings. A weaker footprint extends across evidence units, which may contain related facts. Removing identical QA preserves this signal, locating it in shared content rather than repeated records.

5 Conclusion

We introduced ancestral membership auditing to study whether source participation remains detectable when only synthetic descendants enter downstream training. Across Enron and BillSum, regenerated siblings reveal a source-level signal that persists beyond exact training-record recurrence and under generator mismatch. These results suggest that synthetic transformation can change training records without necessarily erasing evidence of their source provenance. Our study focuses on provenance depth $n = 1$ and controlled QA fine-tuning on two domains, with one training realization per setting. Future work can examine whether ancestral membership persists across longer provenance chains, other synthetic transformations, mixed training data, and independent retraining runs. Extending the audit to settings with more limited source or model access would further clarify its applicability beyond the current setting.

AI USE STATEMENT

We use AI to refine the sentence structure and handle related tasks, especially for the search of pre-print materials. As part of the experimental methodology, the Qwen, Granite, and Phi generators produced synthetic QA for target training, audit probes, and reference calibration (Section 4.1).

ETHICS STATEMENT

This work studies privacy risks to support authorized auditing of training data use. The reported evaluations concern models fine-tuned for this study on synthetic QA derived from Enron emails and BillSum bill text. Public availability of source data does not eliminate privacy concerns: emails can contain personal information, and synthetic derivatives can retain sensitive content. The manuscript reports aggregate audit results rather than making membership claims about identified individuals. The same methods could be misused to investigate data use without authorization. Applications should therefore require appropriate authorization, and any release of derived data or trained adapters should account for privacy risks and the terms governing the source material. Audit predictions provide statistical evidence of participation, not definitive conclusions about an individual source.

REPRODUCIBILITY STATEMENT

An anonymized code package has been prepared, containing the implementation, configuration files, and step-by-step instructions for data preparation, synthetic QA generation, target and reference fine-tuning, audit scoring, and source-level evaluation. The package is code-only: datasets, generated QA, trained model weights, and experimental results are not included; the instructions describe how to obtain or construct the required inputs.

Section 3 defines the audit objective, access assumptions, and scoring rules. Section 4.1 and Appendix A describe data roles, training and audit budgets, and the shared training protocol. Appendix B specifies baseline adaptations, while Appendix C describes aggregation, source-level resampling, and ROC construction. Diagnostic and sensitivity protocols are detailed in Appendices F and G. Within each domain, comparisons reuse the same frozen source assignments and QA banks across model families. Independent regeneration may produce different QA realizations, so reproducing the protocol does not guarantee identical numerical results.

REFERENCES

Atilla Akkus, Masoud Poorghaffar Aghdam, Mingjie Li, Junjie Chu, Michael Backes, Yuyang Zhang, and Sinem Sav. Generated data with fake privacy: Hidden dangers of fine-tuning large language models on generated data. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 8075–8093, Seattle, WA, August 2025. USENIX Association. ISBN 978-1-939133-52-6. URL https://www.usenix.org/conference/usenixsecurity25/presentation/akkus.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914, 2022. doi: 10.1109/SP46214.2022.9833649.

Yuetian Chen, Yuntao Du, Kaiyuan Zhang, Ashish Kundu, Charles Fleming, Bruno Ribeiro, and Ninghui Li. Window-based membership inference attacks against fine-tuned large language models. In *35th USENIX Security Symposium (USENIX Security 26)*, pp. 1187–1206, Baltimore, MD, August 2026. USENIX Association. ISBN 978-1-939133-58-8. URL https://www.usenix.org/conference/usenixsecurity26/presentation/chen-yuetian.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. Instruction pre-training: Language models are supervised multitask learners. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2529–2550, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.148. URL https://aclanthology.org/2024.emnlp-main.148/.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference

attacks work on large language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=av0D19pSkU.

Jamie Hayes, I Shumailov, Christopher A. Choquette-Choo, Matthew Jagielski, Georgios Kaissis, Milad Nasr, Meenatchi Sundaram Muthu Selva Annamalai, Niloofar Mireshghallah, Igor Shilov, Matthieu Meeus, Yves-Alexandre de Montjoye, Katherine Lee, Franziska Boenisch, Adam Dziedzic, and A. Feder Cooper. Exploring the limits of strong membership inference attacks on large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 87729–87795. Curran Associates, Inc., 2025. doi: 10.52202/085713-2934. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/7f0f3a1d358f8a902b1c4b15a6848d71-Paper-Conference.pdf.

Richard Hohensinner, Belgin Mutlu, Inti Gabriel Mendoza Estrada, Matej Vukovic, Simone Kopeinik, and Roman Kern. Tracing the data trail: A survey of data provenance, transparency and traceability in llms, 2026. URL https://arxiv.org/abs/2601.14311.

David Ilić, David Stanojević, and Kostadin Cvejoski. Powerful training-free membership inference against fine-tuned autoregressive language models. 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. 14077–14093, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.640. URL https://aclanthology.org/2026.acl-long.640/.

Matthew Jagielski, Milad Nasr, Katherine Lee, Christopher A. Choquette-Choo, Nicholas Carlini, and Florian Tramer. Students parrot their teachers: Membership inference on model distillation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 44382–44397. Curran Associates, Inc., 2023. doi: 10.52202/075280-1921. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8b07d224a643b02e7571e083578a86d2-Paper-Conference.pdf.

Nikhil Kandpal, Krishna Pillutla, Alina Oprea, Peter Kairouz, Christopher A. Choquette-Choo, and Zheng Xu. User inference attacks on large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18238–18265, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1014. URL https://aclanthology.org/2024.emnlp-main.1014/.

Shayne Longpre, Robert Mahari, Anthony Chen, Naana Obeng-Marnu, Damien Sileo, William Brannon, Niklas Muennighoff, Nathan Khazam, Jad Kabbara, Kartik Perisetla, Xinyi Wu, Enrico Shippole, Kurt Bollacker, Tongshuang Wu, Luis Villa, Sandy Pentland, and Sara Hooker. A large-scale audit of dataset licensing and attribution in AI. *Nature Machine Intelligence*, 6(8): 975–987, August 2024. doi: 10.1038/s42256-024-00878-8. URL https://www.nature.com/articles/s42256-024-00878-8.

Zhexi Lu, Hongliang Chi, Nathalie Baracaldo, Swanand Ravindra Kadhe, Yuseok Jeon, and Lei Yu 0002. In-context probing for membership inference in fine-tuned language models. In *33rd Annual Network and Distributed System Security Symposium, NDSS 2026, San Diego, California, USA, February 23-27, 2026*. The Internet Society, 2026. URL https://www.ndss-symposium.org/ndss-paper/in-context-probing-for-membership-inference-in-fine-tuned-language-models/.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset? 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. 124069–124092. Curran Associates, Inc., 2024. doi: 10.52202/079017-3941. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e01519b47118e2f51aa643151350c905-Paper-Conference.pdf.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.719. URL https://aclanthology.org/2023.findings-acl.719/.

Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. Did the neurons read your book? document-level membership inference for large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 2369–2385, Philadelphia, PA, August 2024. USENIX Association. ISBN 978-1-939133-44-1. URL https://www.usenix.org/conference/usenixsecurity24/presentation/meeus.

Matthieu Meeus, Lukas Wutschitz, Santiago Zanella-Beguelin, Shruti Tople, and Reza Shokri. The canary's echo: Auditing privacy risks of LLM-generated synthetic text. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43557–43580. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/meeus25a.html.

Ali Naseh and Niloofar Mireshghallah. Synthetic data can mislead evaluations: Membership inference as machine text detection, 2025. URL https://arxiv.org/abs/2501.11786.

Nihal Nayak, Yiyang Nan, Avi Trost, and Stephen Bach. Learning to generate instruction tuning datasets for zero-shot task adaptation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12585–12611, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.748. URL https://aclanthology.org/2024.findings-acl.748/.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 51826–51843, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/e32ad85fa27be4a9868d55703f01323e-Paper-Conference.pdf.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017. doi: 10.1109/SP.2017.41.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. 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. 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

Yixin Wu, Ziqing Yang, Yun Shen, Michael Backes, and Yang Zhang. Synthetic artifact auditing: Tracing LLM-Generated synthetic data usage in downstream applications. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 1689–1708, Seattle, WA, August 2025. USENIX Association. ISBN 978-1-939133-52-6. URL https://www.usenix.org/conference/usenixsecurity25/presentation/wu-yixin-auditing.

Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58244–58282. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zarifzadeh24a.html.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Yang, and Hai Li. Min-k%++: Improved baseline for pre-training data detection from large language models.

In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 64845–64862, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a2e3b4132ab2e0b7a21e6e75da7f91a9-Paper-Conference.pdf.

A Protocol and Implementation Details

A.1 Sources, data roles, and synthesis

Each dataset contains 1,000 evaluation sources, split into 500 members and 500 nonmembers, and five mutually disjoint auxiliary cohorts of 500 OUT sources. All OUT cohorts are disjoint from the evaluation cohort. Source identities, membership assignments, and QA banks are reused across target families within a dataset. Auxiliary cohort identities are also shared across generators; the generated records differ with the generator. Enron sources are cleaned bodies of the current messages and BillSum sources are bill text. Neither raw source context nor provenance/evidence metadata is appended to target or reference training inputs.

The provider/auditor generators are Qwen3-4B-Instruct-2507, Granite-4.1-8B, and Phi-4-mini-instruct. Target/base families are DeepSeek-Coder, Qwen3, SmolLM2, OLMo2, RedPajama, OpenLLaMA, Falcon, and Pythia. Each dataset has three targets and fifteen references per family: 24 targets and 120 references in total. Targets are reused across auditors, yielding 72 primary evaluation configurations per domain rather than 72 independent training replications.

A.2 Generation and serialization

QA records are generated in separate calls conditioned on the source. Before assigning training and audit roles, selection prioritizes exact QA diversity without using those roles; repetitions are retained where necessary. The practical audit therefore retains natural recurrence, and selected positions are not assumed IID. Removing audit QA that exactly matches actual training records is a separate analysis performed by the researcher. Inputs use the fixed Instruction/Response serialization. Training supervises the answer and EOS; canonical scoring uses only answer content tokens. Each QA is normalized by its own answer length before source-level averaging.

A.3 Training configuration

Targets and references share the nominal configuration in Table A1. References train on 2,000 QA records, as do targets in the main comparison. LoRA module placement depends on the architecture. The final epoch is fixed for the main comparison; epoch analyses do not select a replacement checkpoint.

Table A1: Shared target/reference fine-tuning configuration.

Parameter	Value
QA records per model	2,000
Epochs	5
LoRA rank / alpha	16 / 32
LoRA dropout	0.05
Learning rate	2×10^{-4}
Effective batch size	8
Precision	BF16
Quantization	None
Training seed	42

B Scoring and Baseline Adaptations

B.1 Canonical scores and sequence UserInf

Canonical scores normalize each QA by the number of answer tokens, exclude EOS, and average QA scores equally within each source. The sequence UserInf adaptation instead retains weighting by answer length:

$$\mathsf{S}_{\mathrm{UI}}(s) = \frac{1}{L}\sum_{j=1}^{L} |a_j|[\ell_B(z_j) - \ell_T(z_j)], \qquad \mathsf{S}_{\mathrm{UI,norm}}(s) = \mathsf{S}_{\mathrm{MSG}}(s). \tag{6}$$

The equality follows under the same scoring tokens and source averaging. It does not make the sequence baseline weighted by answer length identical to MSG.

B.2 Token and window-based comparisons

Min-K% and Min-K%++ use the fixed 20% reporting point. WBC (Chen et al., 2026) uses window lengths 2, 3, 4, 6, 9, 13, 18, 25, 32, and 40, strict local sign votes, and equal QA/source averaging. EZ-MIA (Ilić et al., 2026) computes the Error Zone score for each QA and preserves the original rules for insufficient errors and sentinel values before averaging within each source. WBC uses the authors' core implementation; EZ-MIA uses the authors' score for each record followed by averaging. The alternative that pools records across the sibling family is not used in the main table.

B.3 Offline source-level RMIA

The offline RMIA adaptation uses $a = 1$ and $\gamma = 1$. Query normalization uses five OUT references trained using the auditor's generator, and population records use the four references remaining after leaving out their owning reference. Reference averaging is in probability space, unlike Multi5's arithmetic mean in NLL space. Results for individual QA pairs are averaged within each source. The Enron population contains 2,500 OUT sources and 10,000 auxiliary QA per auditor, with no evaluation nonmembers in the population.

B.4 Access and resource accounting

All principal comparisons use the same bank of probes for the target, but their auxiliary requirements differ. Target-only and readouts based on token probabilities use target outputs, whereas UserInf/MSG use the pretrained base and Multi5 uses fine-tuned references. RMIA additionally scores auxiliary population records. Evaluations of individual records under the target, local auxiliary scoring, QA generation, reference training, and population scoring are separate cost components. Batch calls are not individual record queries. Substituted pretrained bases are not additional fine-tuned references.

C Statistics and Model-Specific Results

C.1 Aggregation and paired uncertainty

Let $M_{i,p,a}$ be a source-level metric for target family i, provider p, and auditor a. For each domain,

$$\overline{M}_{\text{same}} = \frac{1}{24}\sum_{i=1}^{8}\sum_{p=a} M_{i,p,a}, \qquad \overline{M}_{\text{cross}} = \frac{1}{48}\sum_{i=1}^{8}\sum_{p\neq a} M_{i,p,a}. \tag{7}$$

The summary across all nine generator pairs is $(\overline{M}_{\text{same}} + 2\overline{M}_{\text{cross}})/3$. Metrics are computed per configuration; source scores are not pooled across configurations. The two domains have different source cohorts and are not paired with each other.

The Enron paired analysis uses 10,000 stratified bootstrap replicates that resample sources. Each replicate samples 500 members and 500 nonmembers with replacement, preserving complete sibling families and sharing the source resample across methods, generator conditions, and target families. Metrics and their averages over the fixed evaluation scope are recomputed within each replicate. The analysis paired across model families uses seed 20270919; other Enron statistics include analyses with seed 42, and BillSum statistics use seed 42. Intervals retain their original dataset/method/scope. Missing aggregate intervals are not constructed by averaging cellwise endpoints.

With 500 nonmembers, empirical FPR changes in increments of 0.2 percentage points. A 1% budget consequently allows only five false positives. These ROC operating points measured on the test set are distinct from threshold calibration on independent data. Source bootstrap is conditional on the trained models and QA banks; variability across random subsets and retraining variation are different uncertainty sources.

C.2 AUC BY MODEL FAMILY AND GENERATOR ALIGNMENT

Abbreviations: Tgt = Target-only; MK/MK++ = fixed 20% Min-K variants; UI = sequence UserInf; WBC = author core; EZ = author record mean. Each row averages a fixed target family over the specified generator pairs. The comparison uses coverage of all nine generator pairs for each family. ICP-MIA per-family results are in Appendix H.

Table A2: Enron, Same G: per-family AUC averaged over three generator pairs.

Family	Tgt	MK	MK++	UI	WBC	EZ	RMIA	MSG	Multi5
DeepSeek	0.7957	0.8074	0.6930	0.8734	0.8910	0.7565	0.8984	0.8972	0.9306
Qwen3	0.7770	0.7897	0.6302	0.8269	0.8666	0.7000	0.8810	0.8658	0.9121
SmolLM2	0.7673	0.7902	0.7003	0.8509	0.8634	0.7324	0.8930	0.8624	0.9161
OLMo2	0.7900	0.8173	0.6498	0.8473	0.8628	0.6789	0.8809	0.8678	0.9203
RedPajama	0.8063	0.8384	0.6398	0.8484	0.8842	0.6477	0.8971	0.8842	0.9296
OpenLLaMA	0.8046	0.8147	0.6243	0.8665	0.8792	0.6407	0.8945	0.8850	0.9314
Falcon	0.7942	0.8219	0.6713	0.8780	0.8593	0.6399	0.8949	0.8766	0.9276
Pythia	0.7666	0.7892	0.7126	0.8840	0.8397	0.6794	0.8891	0.8518	0.9245

Table A3: Enron, Cross G: per-family AUC averaged over six generator pairs.

Family	Tgt	MK	MK++	UI	WBC	EZ	RMIA	MSG	Multi5
DeepSeek	0.7134	0.7166	0.6180	0.7842	0.8189	0.7306	0.8093	0.8162	0.8447
Qwen3	0.7013	0.7028	0.5802	0.7349	0.7912	0.6734	0.7880	0.7818	0.8213
SmolLM2	0.6928	0.7056	0.6206	0.7582	0.7888	0.6827	0.7940	0.7801	0.8163
OLMo2	0.7050	0.7209	0.5826	0.7514	0.7837	0.6543	0.7853	0.7819	0.8218
RedPajama	0.7231	0.7404	0.5799	0.7542	0.8092	0.6346	0.8033	0.8014	0.8388
OpenLLaMA	0.7175	0.7108	0.5702	0.7754	0.8122	0.6447	0.8060	0.8059	0.8454
Falcon	0.7147	0.7252	0.5997	0.7887	0.7873	0.6386	0.8040	0.7944	0.8424
Pythia	0.6931	0.7070	0.6383	0.7973	0.7743	0.6527	0.8025	0.7750	0.8317

Table A4: BillSum, Same G: per-family AUC averaged over three generator pairs.

Family	Tgt	MK	MK++	UI	WBC	EZ	RMIA	MSG	Multi5
DeepSeek	0.8024	0.8018	0.7000	0.8997	0.8709	0.8142	0.8985	0.8967	0.9523
Qwen3	0.8127	0.8129	0.6675	0.8641	0.8738	0.7560	0.8999	0.8842	0.9472
SmolLM2	0.7902	0.7957	0.7012	0.8825	0.8591	0.7840	0.9003	0.8638	0.9402
OLMo2	0.8075	0.8218	0.7023	0.8843	0.8620	0.7773	0.8960	0.8718	0.9479
RedPajama	0.8280	0.8389	0.6629	0.8703	0.8979	0.6897	0.9095	0.8968	0.9493
OpenLLaMA	0.8284	0.8293	0.6502	0.8842	0.8946	0.6935	0.8906	0.9059	0.9461
Falcon	0.8204	0.8294	0.7128	0.9079	0.8741	0.7542	0.9013	0.8955	0.9513
Pythia	0.7810	0.7967	0.7441	0.9083	0.8542	0.7896	0.8964	0.8580	0.9470

Table A5: BillSum, Cross G: per-family AUC averaged over six generator pairs.

Family	Tgt	MK	MK++	UI	WBC	EZ	RMIA	MSG	Multi5
DeepSeek	0.7016	0.6944	0.6120	0.8048	0.7926	0.7580	0.7997	0.8012	0.8573
Qwen3	0.7065	0.6999	0.5915	0.7551	0.7869	0.7020	0.7976	0.7793	0.8497
SmolLM2	0.6916	0.6905	0.6165	0.7742	0.7775	0.7258	0.7966	0.7682	0.8401
OLMo2	0.7026	0.7067	0.6096	0.7774	0.7846	0.7177	0.7990	0.7761	0.8510
RedPajama	0.7105	0.7045	0.5725	0.7556	0.8099	0.6887	0.8023	0.7924	0.8495
OpenLLaMA	0.7209	0.7101	0.5700	0.7771	0.8146	0.6852	0.8009	0.8019	0.8568
Falcon	0.7123	0.7114	0.6164	0.8085	0.8024	0.7293	0.8082	0.8037	0.8575
Pythia	0.6755	0.6841	0.6383	0.8151	0.7744	0.7320	0.7997	0.7662	0.8414

C.3 MACRO UNCERTAINTY AND PAIRED IMPROVEMENTS

The following intervals accompany the corresponding macro estimates; they are not averages of confidence limits for individual families. For BillSum, the existing 10,000 paired bootstrap draws that resample sources (seed 42) are averaged over the fixed generator conditions and eight families within each draw. The 2.5th and 97.5th percentiles give the reported intervals. Paired differences use Multi5 and MSG from the same draw.

Table A6: AUC and 95% bootstrap intervals by generator condition, with sources as the resampling unit.

Dataset	G condition	Method	AUC	95% CI
Enron	Same G	Target-only	0.7877	[0.7667, 0.8083]
Enron	Same G	UserInf	0.8594	[0.8423, 0.8761]
Enron	Same G	RMIA	0.8911	[0.8801, 0.9019]
Enron	Same G	MSG	0.8739	[0.8581, 0.8891]
Enron	Same G	Multi5	0.9240	[0.9135, 0.9334]
Enron	Cross G	Target-only	0.7076	[0.6838, 0.7310]
Enron	Cross G	UserInf	0.7680	[0.7468, 0.7891]
Enron	Cross G	RMIA	0.7991	[0.7833, 0.8146]
Enron	Cross G	MSG	0.7921	[0.7735, 0.8102]
Enron	Cross G	Multi5	0.8328	[0.8184, 0.8466]
BillSum	Same G	Target-only	0.8088	[0.7907, 0.8265]
BillSum	Same G	UserInf	0.8877	[0.8744, 0.9006]
BillSum	Same G	RMIA	0.8991	[0.8875, 0.9105]
BillSum	Same G	MSG	0.8841	[0.8707, 0.8975]
BillSum	Same G	Multi5	0.9477	[0.9391, 0.9556]
BillSum	Cross G	Target-only	0.7027	[0.6810, 0.7242]
BillSum	Cross G	UserInf	0.7835	[0.7647, 0.8020]
BillSum	Cross G	RMIA	0.8005	[0.7846, 0.8165]
BillSum	Cross G	MSG	0.7861	[0.7689, 0.8038]
BillSum	Cross G	Multi5	0.8504	[0.8386, 0.8621]

Table A7: Differences in AUC between Multi5 and MSG with paired bootstrap intervals from resampling sources across the fixed families, separately within each domain.

Dataset	G condition	Difference	95% paired CI
Enron	Same G	0.0502	[0.0375, 0.0629]
Enron	Cross G	0.0407	[0.0220, 0.0589]
BillSum	Same G	0.0636	[0.0516, 0.0756]
BillSum	Cross G	0.0643	[0.0471, 0.0813]

C.4 ADDITIONAL OPERATING POINTS AT LOW FPR

C.5 ROC CONSTRUCTION AND COMPLETE READOUTS

For each target family and method, the same-generator ROC averages three separate source-level ROCs at common FPRs. Each component uses its full score ordering. The mean curve is constructed from the union of FPR breakpoints, retaining lower/upper TPR values at vertical segments; scores are not pooled and their direction is not inverted based on labels. AUC in tables remains the arithmetic mean of component AUCs. The display may use sparse markers, but the connecting line retains the complete macro curve.

C.6 RESULTS FOR ORDERED PAIRS OF GENERATORS

The following numerical matrices accompany Figure 4. Rows are providers and columns auditors; every entry averages eight target families. Both directions of each generator pair are retained.

Table A8: TPR at 5% FPR for all primary readouts; aggregation matches Table 1. Values are fractions.

	Enron		BillSum	
Method	Same G	Cross G	Same G	Cross G
Target-only	0.3067	0.1939	0.3344	0.1805
Min-K% (20%)	0.3200	0.2007	0.3479	0.1640
Min-K%++ (20%)	0.2322	0.1489	0.2526	0.1406
UserInf	0.5331	0.3554	0.6088	0.3610
WBC	0.3920	0.2789	0.3974	0.2472
EZ-MIA	0.1765	0.1429	0.2612	0.1642
RMIA	0.5862	0.4065	0.6121	0.3734
ICP-MIA	0.3312	0.2096	0.3945	0.2243
MSG	0.4485	0.3037	0.4640	0.2597
Multi5	0.6863	0.4892	0.8117	0.5587

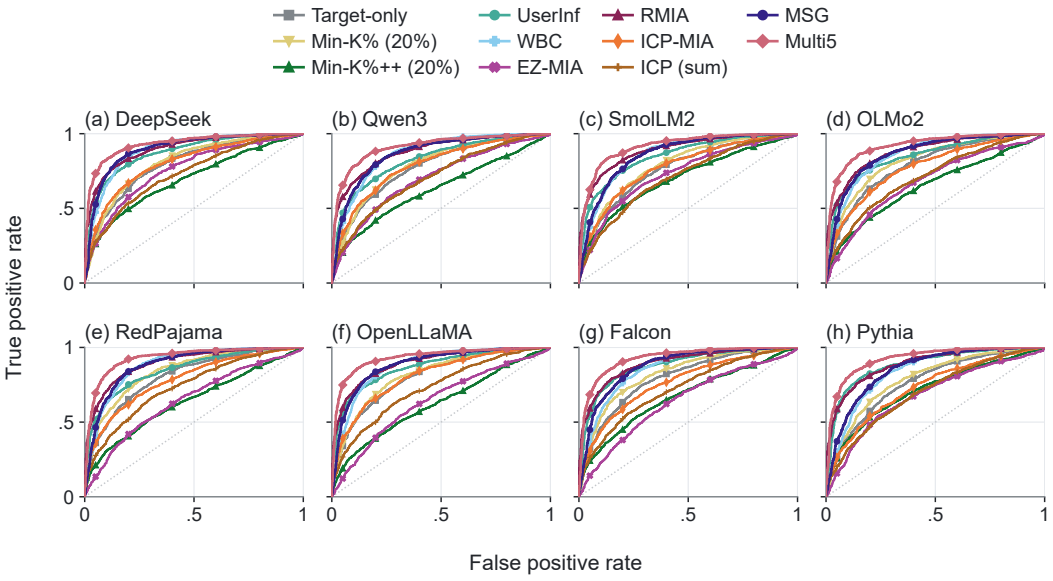


Figure A1: All ten main readouts plus the ICP variant using the sequence sum on Enron under the same generator. Each curve for a target family averages the three matched generator pairs.

Table A9: Enron: AUC for ordered generator pairs, averaged over eight Target families.

Method	Provider	Qwen	Granite	Phi
MSG	Qwen	0.9103	0.8268	0.7863
MSG	Granite	0.8124	0.8805	0.7834
MSG	Phi	0.7637	0.7800	0.8308
Multi5	Qwen	0.9529	0.8785	0.8288
Multi5	Granite	0.8494	0.9321	0.8219
Multi5	Phi	0.7955	0.8227	0.8871

Table A10: BillSum: AUC for ordered generator pairs, averaged over eight Target families.

Method	Provider	Qwen	Granite	Phi
MSG	Qwen	0.9282	0.8355	0.7677
MSG	Granite	0.8287	0.8956	0.7969
MSG	Phi	0.7346	0.7534	0.8285
Multi5	Qwen	0.9716	0.9041	0.8344
Multi5	Granite	0.8893	0.9558	0.8648
Multi5	Phi	0.7790	0.8309	0.9156

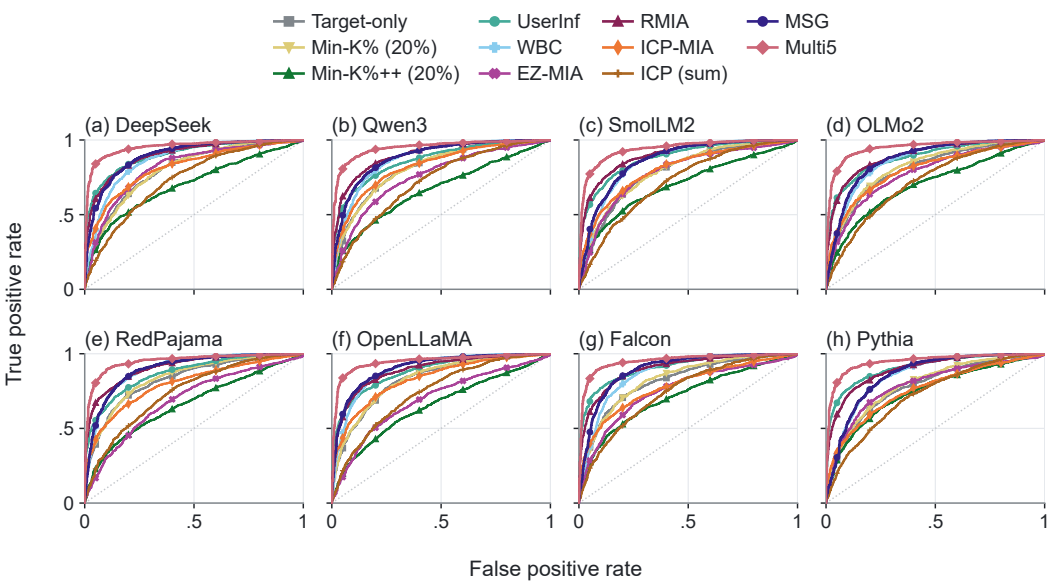


Figure A2: All ten main readouts plus the ICP variant using the sequence sum on BillSum under the same generator. Curve construction and aggregation match Figure 3.

D Heterogeneous Base Evaluation

Let $b_M(z)$ denote the readout in bits per byte of the response used for model M. For auxiliary pretrained base B_j, the source score is

$$\mathrm{S}_{\mathrm{BPB},j}(s) = \frac{1}{L}\sum_{k=1}^{L}[b_{B_j}(z_k) - b_T(z_k)]. \tag{8}$$

Both matched and substituted bases use this scoring unit. In particular, the matrix diagonal is not the MSG score averaged over answer tokens. Each entry averages three same-generator or six cross-generator AUCs for a fixed target/base pair. All auxiliary choices are evaluated without selecting among them using test labels. Model order is DS, Q3, S2, O2, RP, OL, FA, PY: DeepSeek, Qwen3, SmolLM2, OLMo2, RedPajama, OpenLLaMA, Falcon, and Pythia.

Table A11: Enron, Same G: BPB AUC matrix for target and auxiliary base. Rows are Targets; columns are auxiliary Bases. Bold only identifies the diagonal where base families match.

Target	DS	Q3	S2	O2	RP	OL	FA	PY
DeepSeek	**0.8793**	0.8627	0.8509	0.8602	0.8425	0.8631	0.8561	0.8658
Qwen3	0.8512	**0.8545**	0.8409	0.8508	0.8363	0.8549	0.8482	0.8594
SmolLM2	0.8529	0.8502	**0.8463**	0.8542	0.8378	0.8605	0.8517	0.8575
OLMo2	0.8535	0.8525	0.8435	**0.8567**	0.8352	0.8587	0.8511	0.8609
RedPajama	0.8646	0.8726	0.8693	0.8747	**0.8705**	0.8775	0.8725	0.8814
OpenLLaMA	0.8561	0.8613	0.8520	0.8606	0.8481	**0.8709**	0.8618	0.8701
Falcon	0.8522	0.8532	0.8435	0.8564	0.8389	0.8682	**0.8617**	0.8685
Pythia	0.8106	0.8037	0.7875	0.8037	0.7821	0.8153	0.8038	**0.8275**

The off-diagonal mean averages 56 AUCs obtained using individual bases. In contrast, Other7 combines scores from seven other bases before computing AUC. Omitting the matching base uses information about the target family; Other7 is therefore a procedure that requires that information.

Table A12: Enron, Cross G: BPB AUC matrix for target and auxiliary base. Rows are Targets; columns are auxiliary Bases. Bold only identifies the diagonal where base families match.

Target	DS	Q3	S2	O2	RP	OL	FA	PY
DeepSeek	**0.7973**	0.7826	0.7727	0.7790	0.7658	0.7801	0.7737	0.7848
Qwen3	0.7704	**0.7799**	0.7649	0.7725	0.7612	0.7767	0.7689	0.7831
SmolLM2	0.7713	0.7747	**0.7713**	0.7788	0.7685	0.7822	0.7750	0.7808
OLMo2	0.7639	0.7718	0.7621	**0.7756**	0.7572	0.7739	0.7690	0.7830
RedPajama	0.7816	0.7929	0.7870	0.7931	**0.7927**	0.7979	0.7918	0.8010
OpenLLaMA	0.7766	0.7860	0.7757	0.7852	0.7765	**0.7996**	0.7871	0.7953
Falcon	0.7675	0.7746	0.7641	0.7759	0.7650	0.7878	**0.7824**	0.7902
Pythia	0.7369	0.7350	0.7188	0.7351	0.7193	0.7456	0.7351	**0.7597**

Table A13: BillSum, Same G: BPB AUC matrix for target and auxiliary base. Rows are Targets; columns are auxiliary Bases. Bold only identifies the diagonal where base families match.

Target	DS	Q3	S2	O2	RP	OL	FA	PY
DeepSeek	**0.8598**	0.8263	0.8133	0.8257	0.8299	0.8456	0.8434	0.8271
Qwen3	0.8504	**0.8526**	0.8400	0.8513	0.8530	0.8723	0.8657	0.8462
SmolLM2	0.8136	0.8162	**0.8172**	0.8234	0.8229	0.8447	0.8382	0.8156
OLMo2	0.8362	0.8383	0.8352	**0.8481**	0.8459	0.8675	0.8608	0.8397
RedPajama	0.8531	0.8600	0.8556	0.8634	**0.8757**	0.8820	0.8774	0.8653
OpenLLaMA	0.8456	0.8510	0.8477	0.8547	0.8613	**0.8759**	0.8707	0.8527
Falcon	0.8427	0.8398	0.8352	0.8447	0.8525	0.8731	**0.8702**	0.8458
Pythia	0.8150	0.8128	0.8035	0.8181	0.8251	0.8514	0.8445	**0.8229**

Table A14: BillSum, Cross G: BPB AUC matrix for target and auxiliary base. Rows are Targets; columns are auxiliary Bases. Bold only identifies the diagonal where base families match.

Target	DS	Q3	S2	O2	RP	OL	FA	PY
DeepSeek	**0.7661**	0.7367	0.7257	0.7333	0.7362	0.7466	0.7487	0.7368
Qwen3	0.7491	**0.7545**	0.7444	0.7528	0.7536	0.7661	0.7631	0.7501
SmolLM2	0.7262	0.7304	**0.7349**	0.7362	0.7357	0.7495	0.7469	0.7301
OLMo2	0.7401	0.7439	0.7421	**0.7542**	0.7497	0.7656	0.7639	0.7460
RedPajama	0.7460	0.7566	0.7502	0.7589	**0.7736**	0.7719	0.7702	0.7606
OpenLLaMA	0.7495	0.7618	0.7575	0.7644	0.7695	**0.7780**	0.7749	0.7619
Falcon	0.7436	0.7491	0.7490	0.7553	0.7615	0.7762	**0.7785**	0.7572
Pythia	0.7184	0.7224	0.7176	0.7285	0.7349	0.7520	0.7505	**0.7374**

Table A15: Summaries of base substitution under the common BPB readout. Diagonal/off-diagonal entries average AUCs from individual bases; Other7 is a separate ensemble that combines scores. Tgt-only BPB uses no auxiliary base.

Dataset	G condition	Diagonal	Off-diagonal	Other7	Tgt-only BPB
Enron	Same G	0.8584	0.8495	0.8558	0.7322
Enron	Cross G	0.7823	0.7719	0.7787	0.6612
BillSum	Same G	0.8528	0.8435	0.8491	0.7795
BillSum	Cross G	0.7596	0.7486	0.7539	0.6742

E Comparison with Direct Source Scoring

Direct-S uses mean NLL on source text under plain causal scoring without QA/chat serialization. It averages three provider targets and has no auditor generator axis. The same comparator is used alongside both sibling alignment groups. Both Direct-S base gain and MSG use base/target likelihoods, but their input format and token/query budgets differ. The Enron intervals below apply to the AUC difference between sibling MSG and Direct-S when generators match; its cross-generator

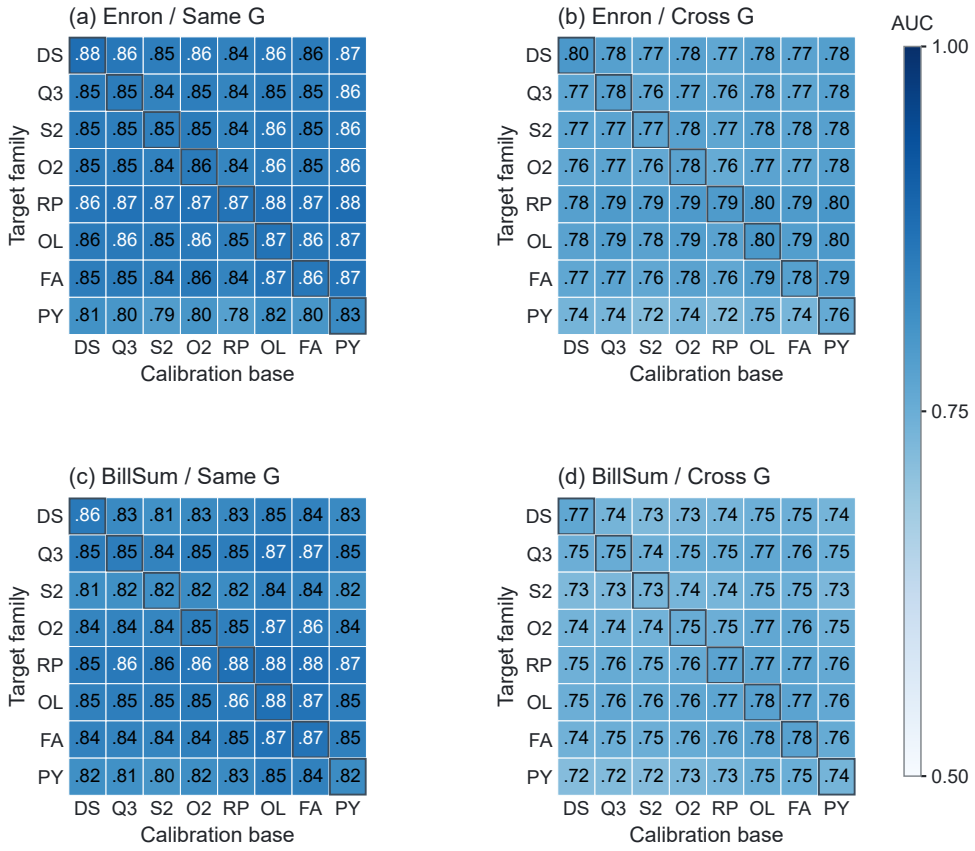


Figure A3: Complete BPB matrices for target and auxiliary base on both domains, separated by generator alignment. All diagonal and off-diagonal entries use BPB. DS: DeepSeek; Q3: Qwen3; S2: SmolLM2; O2: OLMo2; RP: RedPajama; OL: OpenLLaMA; FA: Falcon; PY: Pythia.

values are point estimates. For BillSum, Table A18 supplies intervals for paired differences for both alignment conditions, using the same source draws as Direct-S.

Table A16: Enron: direct source scoring versus sibling auditing under base calibration. The paired interval applies to the same-generator AUC difference.

Family	Direct-S	Same G	Cross G	Δ same	95% paired CI
DeepSeek	0.7471	0.8972	0.8162	0.1501	[0.1252, 0.1753]
Qwen3	0.6992	0.8658	0.7818	0.1666	[0.1395, 0.1943]
SmolLM2	0.7290	0.8624	0.7801	0.1333	[0.1055, 0.1623]
OLMo2	0.7365	0.8678	0.7819	0.1313	[0.1021, 0.1605]
RedPajama	0.7232	0.8842	0.8014	0.1609	[0.1317, 0.1908]
OpenLLaMA	0.7799	0.8850	0.8059	0.1051	[0.0807, 0.1305]
Falcon	0.7152	0.8766	0.7944	0.1614	[0.1320, 0.1920]
Pythia	0.7195	0.8518	0.7750	0.1323	[0.0990, 0.1652]

Table A17: BillSum Direct-S and sibling MSG. Values are per-family means; intervals for paired differences are in Table A18.

Family	Direct-S	Same G	Cross G	Δ same	Δ cross
DeepSeek	0.7502	0.8967	0.8012	0.1465	0.0510
Qwen3	0.7661	0.8842	0.7793	0.1182	0.0132
SmolLM2	0.7282	0.8638	0.7682	0.1356	0.0400
OLMo2	0.7359	0.8718	0.7761	0.1359	0.0401
RedPajama	0.7270	0.8968	0.7924	0.1699	0.0655
OpenLLaMA	0.7590	0.9059	0.8019	0.1469	0.0429
Falcon	0.6853	0.8955	0.8037	0.2102	0.1183
Pythia	0.7459	0.8580	0.7662	0.1120	0.0203

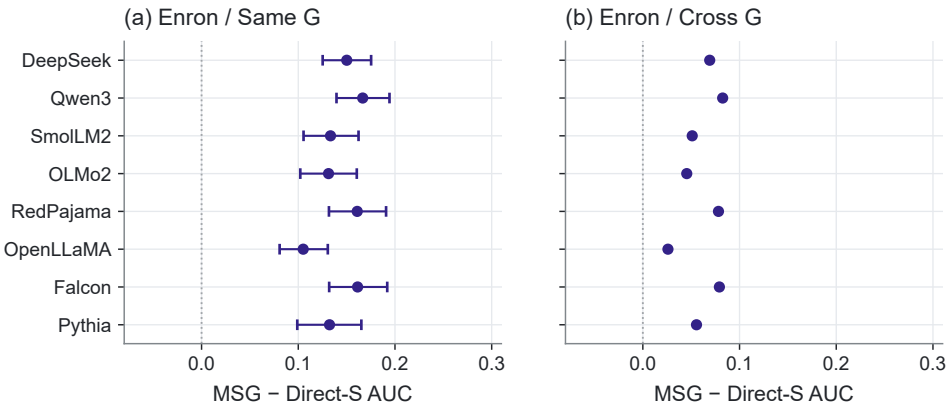


Figure A4: Per-family improvement of sibling MSG over Direct-S on Enron, separated by generator alignment. Same-generator error bars are saved paired bootstrap intervals from resampling sources; cross-generator values are point estimates only. The comparison uses the same base and target models but different probe formats and budgets.

Table A18: BillSum: paired source-bootstrap intervals for sibling MSG minus Direct-S gain, computed separately for same and cross generator conditions.

Family	Δ same	95% paired CI	Δ cross	95% paired CI
DeepSeek	0.1465	[0.1199, 0.1739]	0.0510	[0.0226, 0.0797]
Qwen3	0.1182	[0.0932, 0.1436]	0.0132	[-0.0133, 0.0398]
SmolLM2	0.1356	[0.1080, 0.1629]	0.0400	[0.0116, 0.0682]
OLMo2	0.1359	[0.1069, 0.1652]	0.0401	[0.0100, 0.0708]
RedPajama	0.1699	[0.1415, 0.1991]	0.0655	[0.0363, 0.0956]
OpenLLaMA	0.1469	[0.1220, 0.1724]	0.0429	[0.0176, 0.0690]
Falcon	0.2102	[0.1799, 0.2415]	0.1183	[0.0874, 0.1503]
Pythia	0.1120	[0.0843, 0.1408]	0.0203	[-0.0094, 0.0505]

F Recurrence, Genealogy, and Evidence Diagnostics

F.1 Controls for identical records and genealogy

The oracle filter removes audit QA that exactly matches records in the target's actual training set and recomputes scores over the retained siblings. It requires knowledge of hidden training records and is not part of the practical auditor. Filtering changes probe count and composition; the AUC difference before and after filtering is not a causal contribution of repetition. Provider balancing during filtering and removal of duplicates within the audit bank are separate controls. Genealogy shuffling permutes whole sibling families without fixed points, leaving each family's internal structure intact.

Table A19: MSG under natural regeneration, oracle removal of exact matches to training records, and genealogy shuffling that keeps sibling families intact. Entries are mean source-level AUCs.

Dataset	G condition	Natural	Exact matches removed	Shuffled
Enron	Same G	0.8739	0.8702	0.5000
Enron	Cross G	0.7921	0.7904	0.5002
BillSum	Same G	0.8841	0.8828	0.4999
BillSum	Cross G	0.7861	0.7857	0.5000

F.2 Adaptation gains and sign patterns

The target and reference gains relative to the pretrained base are

$$g_T(s) = \overline{\ell}_B(s) - \overline{\ell}_T(s), \tag{9}$$

$$g_R(s) = \overline{\ell}_B(s) - \frac{1}{5}\sum_{i=1}^{5} \overline{\ell}_{R_{A,i}}(s). \tag{10}$$

Thus $\mathsf{S}_{\mathrm{Multi5}}(s) = g_T(s) - g_R(s)$. Positive member mean gain and relative retention are distinguished by the signs of the member/nonmember group means, as in Section 4.7.1. Counts summarize these signs across configurations; they are not significance tests for individual sources. Gain magnitudes across different tokenizers are not used as a common leakage scale.

Table A20: Adaptation patterns in group means. Positive member gain indicates absolute improvement; relative retention has both groups below zero with a higher member mean. Counts are descriptive configurations.

Dataset	G condition	Positive member gain	Relative retention
Enron	Same G	21	3
Enron	Cross G	20	28
BillSum	Same G	21	3
BillSum	Cross G	16	32

F.3 Common support and equal QA budgets

Evidence annotations associate audit siblings with examples generated for the provider. Provider examples from nonmember sources supply diagnostic comparators and are not target training records. The comparison uses the same eligible sources for both evidence strata, each excluding exact matches to provider examples. Comparisons with equal QA budgets form a separate control. Distinct evidence units are not assumed to contain semantically independent facts.

Table A21: Multi5 evidence diagnostics. CS uses the same eligible sources in both strata; EQ equalizes QA budgets. S/D denote shared/different evidence, both excluding exact matches to provider examples. CS sources gives the range of eligible source counts across configurations.

Dataset	G condition	CS-S	CS-D	EQ-S	EQ-D	CS sources
Enron	Same G	0.9364	0.5943	0.8780	0.5916	794–874
Enron	Cross G	0.8511	0.5846	0.8033	0.5795	816–882
BillSum	Same G	0.9577	0.6819	0.9512	0.6657	997–999
BillSum	Cross G	0.8719	0.6710	0.8682	0.6394	977–993

G SENSITIVITY ANALYSES

G.1 RANDOM NESTED AUDIT SUBSETS

Each curve uses 500 nested random subsets of the frozen audit bank with 16 positions, without retraining or generating new QA. The tabled means are not the control using the first L positions. In both domains, the aggregate P05–P95 ranges over QA subsets give the central 90% variability band. BillSum bands average the saved metrics within each replicate over the fixed cells and families before taking quantiles. These ranges differ from bootstrap confidence intervals from resampling sources. Zero subset variation at $L = 16$ reflects the single full subset, not zero uncertainty from sampling sources.

Table A22: Enron: AUC over 500 random nested subsets.

	Same G		Cross G	
L	MSG	Multi5	MSG	Multi5
1	0.6876	0.6943	0.6371	0.6392
2	0.7447	0.7659	0.6807	0.6923
4	0.7987	0.8341	0.7247	0.7469
6	0.8262	0.8680	0.7483	0.7768
8	0.8431	0.8883	0.7632	0.7958
10	0.8544	0.9017	0.7736	0.8093
12	0.8627	0.9113	0.7814	0.8191
14	0.8690	0.9185	0.7873	0.8267
16	0.8739	0.9240	0.7921	0.8328

Table A23: BillSum: AUC over 500 random nested subsets.

	Same G		Cross G	
L	MSG	Multi5	MSG	Multi5
1	0.6806	0.7042	0.6253	0.6402
2	0.7429	0.7846	0.6685	0.6969
4	0.8018	0.8590	0.7132	0.7566
6	0.8318	0.8939	0.7382	0.7892
8	0.8504	0.9141	0.7544	0.8099
10	0.8629	0.9271	0.7658	0.8244
12	0.8720	0.9361	0.7744	0.8353
14	0.8788	0.9427	0.7809	0.8437
16	0.8841	0.9477	0.7861	0.8504

G.2 REFERENCE IDENTITY AND ENSEMBLE SIZE

R1–R5 denote individual reference identities. The main count curve uses R1 followed by Multi2–Multi5, all from the same fixed prefix. Results for individual references are reported below but are not averaged to replace the endpoint with one reference. No reference is selected by evaluation AUC.

G.3 EPOCH TRAJECTORIES

The same final references calibrate every target epoch in both domains; references are not retrained to match each epoch. Epoch 5 remains the primary checkpoint. The table reports audit AUC, not training loss.

G.4 NUMBER OF TRAINING MEMBERS WITH FIXED REFERENCES

Protocol. This separate ablation uses DeepSeek-Coder on Enron, not an average over eight families. Fixed nested prefixes of the original 500 member sources give $m \in \{5, 25, 50, 100, 250, 500\}$, or

Table A24: AUC for individual references and ensembles using fixed prefixes. Same/cross conditions are summarized separately for each domain.

	Enron		BillSum	
Reference	Same G	Cross G	Same G	Cross G
R1	0.8881	0.7922	0.9212	0.8219
R2	0.9005	0.8082	0.9199	0.8198
R3	0.8918	0.7964	0.9199	0.8165
R4	0.8874	0.7909	0.9239	0.8224
R5	0.8930	0.7975	0.9173	0.8097
Multi2	0.9129	0.8215	0.9362	0.8386
Multi3	0.9202	0.8287	0.9429	0.8452
Multi4	0.9215	0.8302	0.9463	0.8499
Multi5	0.9240	0.8328	0.9477	0.8504

Table A25: Auditing at successive epochs in both domains, averaged over the fixed families and specified generator pairs.

Dataset	G / readout	E1	E2	E3	E4	E5
Enron	Same / Target-only	0.5846	0.6822	0.7517	0.7810	0.7877
Enron	Same / MSG	0.6307	0.7713	0.8446	0.8708	0.8739
Enron	Same / Multi5	0.5745	0.7325	0.8507	0.9118	0.9240
Enron	Cross / Target-only	0.5640	0.6325	0.6822	0.7033	0.7076
Enron	Cross / MSG	0.5956	0.7089	0.7721	0.7924	0.7921
Enron	Cross / Multi5	0.5411	0.6629	0.7654	0.8214	0.8328
BillSum	Same / Target-only	0.5623	0.6808	0.7661	0.8013	0.8088
BillSum	Same / MSG	0.6209	0.7757	0.8542	0.8819	0.8841
BillSum	Same / Multi5	0.6008	0.7699	0.8883	0.9379	0.9477
BillSum	Cross / Target-only	0.5383	0.6145	0.6732	0.6975	0.7027
BillSum	Cross / MSG	0.5826	0.7012	0.7671	0.7883	0.7861
BillSum	Cross / Multi5	0.5631	0.6785	0.7840	0.8388	0.8504

$1\%, 5\%, 10\%, 20\%, 50\%, 100\%$ of that original cohort. Each source retains its four training QA and sixteen audit QA. Each evaluation includes the selected m members and the same original 500 nonmembers; unselected original members are excluded, not relabeled. Thus $m/500$ is the retained fraction of the original training members, not the membership prevalence of the audit set. Natural QA recurrence is preserved, training includes answer tokens and EOS, and evaluation excludes EOS.

Three provider targets are trained at each of the five smaller scales, with five epochs and the original configuration. The three $m = 500$ targets are reused, giving fifteen additional target training runs and no new reference training. The six scales contain $20, 100, 200, 400, 1000, 2000$ training QA per target, respectively, and use $15, 65, 125, 250, 625, 1250$ optimizer steps. All comparisons retain the original five references trained with the auditor's generator, each trained on 500 disjoint sources. Their saved losses and the base losses are reused.

Statistics and interpretation. Same G and Cross G average the three and six corresponding cell metrics, respectively, rather than pooling predictions. Existing stratified bootstrap intervals use 10,000 replicates and seed 42, resampling m positive sources and 500 negative sources within each scale and binding source multiplicities across methods and cells. They are conditional on the trained models; no significance test comparing scales or method differences is added. With five members, each cell's TPR changes in increments of 20 percentage points, so its results at low FPR are exploratory.

MSG remains discriminative across the tested scales (Figure 5(g,h)). At $m = 25$, its AUCs with the same and different generators are $0.9538/0.8402$, compared with Multi5's $0.8291/0.6969$. At $m = 500$, Multi5 reaches $0.9306/0.8447$, compared with MSG's $0.8972/0.8162$. Multi5 AUC is lower than MSG at $m = 5, 25, 50$ and higher in point estimates at $m = 100, 250, 500$, under both generator conditions. The sweep therefore does not support a calibration benefit independent of training scale. Training data, the composition of participating sources, and optimizer steps change together, while references stay fixed. These results cannot isolate a causal effect of membership

density or establish reference mismatch as the cause; the primary setting with 500 members is unchanged. Tables A26 and A27 report all prescribed scales, including AUC, TPR at 5% and 1% FPR, and the saved intervals.

Table A26: DeepSeek–Enron: varying the number of training members, with the same generator. m is the number of training member sources; the negative set contains 500 fixed sources. Each entry is a point estimate with its saved 95% confidence interval from resampling sources. TPR_5 and TPR_1 use 5% and 1% FPR. Results with five members are exploratory.

m	Method	AUC [95% CI]	TPR_5 [95% CI]	TPR_1 [95% CI]
5	Target-only	0.7225 [0.6105, 0.8021]	0.0667 [0.0000, 0.2000]	0.0000 [0.0000, 0.0000]
	MSG	0.9136 [0.8804, 0.9544]	0.6000 [0.3333, 0.8000]	0.3333 [0.1333, 0.6000]
	Multi5	0.5425 [0.3715, 0.7123]	0.0667 [0.0000, 0.2000]	0.0000 [0.0000, 0.0000]
25	Target-only	0.7993 [0.7370, 0.8558]	0.3200 [0.2000, 0.4667]	0.1600 [0.0667, 0.2667]
	MSG	0.9538 [0.9275, 0.9753]	0.8000 [0.6933, 0.8933]	0.4933 [0.3733, 0.6800]
	Multi5	0.8291 [0.7691, 0.8850]	0.4667 [0.3333, 0.6000]	0.2667 [0.1467, 0.4267]
50	Target-only	0.8056 [0.7590, 0.8494]	0.4000 [0.2600, 0.5000]	0.2600 [0.1400, 0.3467]
	MSG	0.9390 [0.9129, 0.9620]	0.7800 [0.7267, 0.8667]	0.6667 [0.5600, 0.7667]
	Multi5	0.9133 [0.8815, 0.9420]	0.7333 [0.6600, 0.8068]	0.5533 [0.4600, 0.6533]
100	Target-only	0.8185 [0.7832, 0.8505]	0.3867 [0.3067, 0.4600]	0.2433 [0.1633, 0.3133]
	MSG	0.9416 [0.9241, 0.9570]	0.7800 [0.7133, 0.8333]	0.5167 [0.3700, 0.6467]
	Multi5	0.9475 [0.9302, 0.9630]	0.7933 [0.7433, 0.8467]	0.6500 [0.5633, 0.7267]
250	Target-only	0.8129 [0.7886, 0.8365]	0.3627 [0.3040, 0.4400]	0.1360 [0.0853, 0.2133]
	MSG	0.9192 [0.9040, 0.9336]	0.6600 [0.5507, 0.7374]	0.1773 [0.1093, 0.3733]
	Multi5	0.9356 [0.9225, 0.9480]	0.7733 [0.7253, 0.8213]	0.5800 [0.5200, 0.6387]
500	Target-only	0.7957 [0.7735, 0.8169]	0.2993 [0.2220, 0.3753]	0.0913 [0.0453, 0.1480]
	MSG	0.8972 [0.8811, 0.9123]	0.5267 [0.4280, 0.6220]	0.0867 [0.0473, 0.2640]
	Multi5	0.9306 [0.9193, 0.9412]	0.7340 [0.6780, 0.7780]	0.4000 [0.1966, 0.5680]

Table A27: DeepSeek–Enron: varying the number of training members, with different generators. m is the number of training member sources; the negative set contains 500 fixed sources. Each entry is a point estimate with its saved 95% confidence interval from resampling sources. TPR_5 and TPR_1 use 5% and 1% FPR. Results with five members are exploratory.

m	Method	AUC [95% CI]	TPR_5 [95% CI]	TPR_1 [95% CI]
5	Target-only	0.7108 [0.6059, 0.7817]	0.0000 [0.0000, 0.1333]	0.0000 [0.0000, 0.0000]
	MSG	0.9012 [0.8360, 0.9601]	0.6000 [0.3333, 0.8333]	0.3333 [0.1000, 0.6667]
	Multi5	0.5227 [0.3225, 0.7243]	0.0667 [0.0000, 0.2000]	0.0000 [0.0000, 0.0000]
25	Target-only	0.7265 [0.6474, 0.7985]	0.1933 [0.0933, 0.3333]	0.0733 [0.0133, 0.1533]
	MSG	0.8402 [0.7853, 0.8878]	0.4667 [0.3467, 0.6067]	0.3133 [0.1933, 0.4467]
	Multi5	0.6969 [0.6238, 0.7712]	0.3133 [0.1800, 0.4467]	0.1200 [0.0600, 0.2533]
50	Target-only	0.7383 [0.6842, 0.7911]	0.2700 [0.1767, 0.3667]	0.0933 [0.0500, 0.1667]
	MSG	0.8801 [0.8439, 0.9117]	0.6000 [0.5233, 0.6900]	0.4033 [0.3000, 0.4967]
	Multi5	0.8332 [0.7907, 0.8727]	0.5067 [0.4167, 0.5967]	0.3433 [0.2333, 0.4400]
100	Target-only	0.7462 [0.7063, 0.7836]	0.2567 [0.1933, 0.3367]	0.1050 [0.0617, 0.1717]
	MSG	0.8700 [0.8442, 0.8937]	0.5667 [0.4867, 0.6317]	0.3350 [0.2667, 0.4267]
	Multi5	0.8779 [0.8521, 0.9022]	0.6483 [0.5833, 0.7050]	0.4550 [0.3367, 0.5283]
250	Target-only	0.7423 [0.7130, 0.7702]	0.2453 [0.1920, 0.3013]	0.0920 [0.0673, 0.1413]
	MSG	0.8532 [0.8331, 0.8718]	0.4360 [0.3607, 0.5100]	0.2060 [0.1507, 0.2687]
	Multi5	0.8659 [0.8459, 0.8849]	0.5987 [0.5533, 0.6480]	0.4260 [0.3453, 0.4880]
500	Target-only	0.7134 [0.6889, 0.7378]	0.1950 [0.1520, 0.2463]	0.0733 [0.0427, 0.0993]
	MSG	0.8162 [0.7963, 0.8350]	0.3383 [0.2683, 0.4073]	0.0963 [0.0537, 0.1543]
	Multi5	0.8447 [0.8296, 0.8597]	0.5257 [0.4717, 0.5703]	0.2460 [0.1703, 0.3327]

G.5 Additional scoring sensitivities

These secondary analyses reuse the frozen QA banks and trained models. Each reported AUC is computed within a configuration and then averaged over the same eight families and either three same-generator or six cross-generator cells. They do not change the primary readout based on mean gain with EOS excluded, token fraction, audit budget, or checkpoint.

EOS and family normalization. Including EOS changes only the scoring tokens, not training. For a given calibration gain $g_1, \ldots, g_L$, family studentization uses $\overline{g}/(\mathrm{std}_{\mathrm{pop}}(g) + 10^{-6})$. It changes source ranking and is not treated as the canonical transfer gap. Its results are reported alongside the unnormalized score.

Table A28: EOS inclusion at scoring time; training is unchanged. AUCs average the same eight families and generator conditions.

	Enron		BillSum	
Readout	Same G	Cross G	Same G	Cross G
Target-only, no EOS	0.7877	0.7076	0.8088	0.7027
Target-only, with EOS	0.8015	0.7192	0.8107	0.7012
MSG, no EOS	0.8739	0.7921	0.8841	0.7861
MSG, with EOS	0.8285	0.7505	0.8293	0.7385
Multi5, no EOS	0.9240	0.8328	0.9477	0.8504
Multi5, with EOS	0.9256	0.8349	0.9492	0.8486

Table A29: Family studentization of each indicated calibration gain. These are secondary sensitivity results, not replacements for the fixed mean score.

	Enron		BillSum	
Readout	Same G	Cross G	Same G	Cross G
MSG	0.8739	0.7921	0.8841	0.7861
MSG / family SD	0.8774	0.7983	0.8890	0.7956
R1	0.8881	0.7922	0.9212	0.8219
R1 / family SD	0.8554	0.7793	0.8911	0.8166
Multi5	0.9240	0.8328	0.9477	0.8504
Multi5 / family SD	0.8862	0.8158	0.9086	0.8412

Sibling aggregation. The controls retaining the largest gains average the largest indicated fraction of the 16 gains; trimming removes the smallest and largest gain. The smooth maximum readout is $\tau \log[L^{-1} \sum_j \exp(g_j/\tau)]$. All listed temperatures and aggregations are shown, without choosing a replacement based on evaluation AUC.

Token fraction. The token-level Min-K% and Min-K%++ controls retain their original definitions and equal sibling averaging. The full saved 10–100% grid is reported below; the main table remains at the fixed 20% point.

Fixed prefix and random single sibling. The fixed-prefix control uses the first L positions of the frozen descendant order. It is distinct from the curve based on randomly sampled nested subsets. The control using a randomly selected single sibling averages AUC over 100 saved draws within each cell before averaging cells and families; it does not define a bootstrap interval from resampling sources.

H In-Context Probing Baseline

ICP-MIA (Lu et al., 2026) is evaluated on all eight families and all nine pairs of provider and auditor generators in both domains. The main table uses normalization by answer token count from the

Table A30: Alternative sibling aggregations of pretrained-base gain, with the fixed MSG shown for comparison. All use the same 16-position probe bank.

	Enron		BillSum	
Readout	Same G	Cross G	Same G	Cross G
MSG (mean)	0.8739	0.7921	0.8841	0.7861
Median	0.8421	0.7618	0.8454	0.7548
Minimum	0.6394	0.5997	0.6470	0.5929
Maximum	0.8123	0.7447	0.8332	0.7459
Trim one at each end	0.8715	0.7893	0.8824	0.7845
Top 25%	0.8522	0.7771	0.8744	0.7819
Top 50%	0.8700	0.7915	0.8863	0.7925
Top 75%	0.8756	0.7970	0.8874	0.7927
LSE, $\tau = 0.5$	0.8242	0.7562	0.8568	0.7703
LSE, $\tau = 1$	0.8400	0.7697	0.8721	0.7833
LSE, $\tau = 2$	0.8592	0.7852	0.8823	0.7894
Sibling consistency	0.8561	0.7782	0.8576	0.7749
Signed-rank gain	0.8765	0.7973	0.8874	0.7968

Table A31: Token-fraction sensitivity. The main comparison keeps 20%; this table does not select a fraction by evaluation AUC.

	Enron		BillSum	
Readout	Same G	Cross G	Same G	Cross G
Min-K%, 10%	0.7782	0.6862	0.7844	0.6735
Min-K%, 20%	0.8086	0.7162	0.8158	0.7002
Min-K%, 30%	0.8146	0.7250	0.8229	0.7089
Min-K%, 40%	0.8110	0.7244	0.8217	0.7101
Min-K%, 50%	0.8057	0.7214	0.8173	0.7092
Min-K%, 60%	0.7963	0.7135	0.8118	0.7049
Min-K%, 70%	0.7932	0.7114	0.8085	0.7018
Min-K%, 80%	0.7858	0.7048	0.8057	0.7000
Min-K%, 90%	0.7760	0.6967	0.8011	0.6962
Min-K%, 100%	0.7877	0.7076	0.8088	0.7027
Min-K%++, 10%	0.6298	0.5721	0.6592	0.5826
Min-K%++, 20%	0.6651	0.5987	0.6926	0.6034
Min-K%++, 30%	0.6871	0.6166	0.7102	0.6164
Min-K%++, 40%	0.7022	0.6293	0.7199	0.6240
Min-K%++, 50%	0.7127	0.6391	0.7245	0.6288
Min-K%++, 60%	0.7154	0.6418	0.7249	0.6291
Min-K%++, 70%	0.7200	0.6454	0.7271	0.6306
Min-K%++, 80%	0.7208	0.6461	0.7281	0.6316
Min-K%++, 90%	0.7211	0.6465	0.7274	0.6313
Min-K%++, 100%	0.7258	0.6505	0.7311	0.6339

released implementation. The formulation using the sequence sum is retained here as a separate variant; neither changes the generated QA, trained targets, or source labels.

For each audit QA $z = (q, a)$, the fixed adaptation using self-perturbation by word masking constructs 20 demonstration probes by replacing 50% of eligible words with the literal placeholder `word`. The masked demonstration precedes the original query. Only the final query's original answer content tokens are scored; the demonstration, question, padding, and EOS are excluded. No reference model or generation model is used. Let c_j be probe j, $n = |a|$, and $\ell_T(z \mid c_j)$ the corresponding mean

Table A32: Fixed first-L audit positions, a separate control from random nested subsets.

		Enron		BillSum	
Readout	L	Same G	Cross G	Same G	Cross G
MSG	1	0.7035	0.6612	0.6933	0.6373
MSG	2	0.7658	0.6993	0.7656	0.6853
MSG	4	0.8092	0.7364	0.8124	0.7237
MSG	6	0.8394	0.7581	0.8433	0.7476
MSG	8	0.8484	0.7659	0.8563	0.7588
MSG	10	0.8600	0.7763	0.8683	0.7698
MSG	12	0.8653	0.7817	0.8744	0.7780
MSG	14	0.8703	0.7886	0.8807	0.7818
MSG	16	0.8739	0.7921	0.8841	0.7861
Multi5	1	0.7067	0.6583	0.7155	0.6535
Multi5	2	0.7839	0.7082	0.8056	0.7154
Multi5	4	0.8418	0.7569	0.8744	0.7691
Multi5	6	0.8755	0.7821	0.9011	0.7947
Multi5	8	0.8918	0.7944	0.9183	0.8136
Multi5	10	0.9062	0.8100	0.9291	0.8254
Multi5	12	0.9132	0.8183	0.9366	0.8354
Multi5	14	0.9195	0.8278	0.9427	0.8410
Multi5	16	0.9240	0.8328	0.9477	0.8504

Table A33: Random single sibling: AUC averaged over 100 draws and then across the fixed generator cells and student-model families.

	Enron		BillSum	
Readout	Same G	Cross G	Same G	Cross G
MSG	0.6875	0.6380	0.6813	0.6263
Multi5	0.6944	0.6401	0.7036	0.6402

answer NLL. The two source scores are

$$
\begin{aligned}
d_j(z) &= \ell_T(z \mid c_j) - \ell_T(z), \\
\mathsf{S}_{\mathrm{ICP,mean}}(s) &= \frac{1}{L} \sum_{z \in Z_{\mathrm{audit}}(s)} \min_{1 \le j \le 20} d_j(z), \\
\mathsf{S}_{\mathrm{ICP,sum}}(s) &= \frac{1}{L} \sum_{z \in Z_{\mathrm{audit}}(s)} |a_z| \min_{1 \le j \le 20} d_j(z).
\end{aligned} \tag{11}
$$

Higher scores indicate stronger evidence of membership. Both variants use the same conditional and unconditioned forward results; the sequence sum additionally weights each QA by its answer length. The probe budget, masking rate, and the convention that higher scores favor membership are fixed, and the minimum is taken over the prescribed probes, not chosen using membership labels. Word masking, scoring only the response, and source aggregation are explicit task adaptations rather than a claim to reproduce every setting of the original benchmark operating on text samples.

Table A34 reports the two normalizations on the complete cohorts of eight model families, with saved paired bootstrap AUC intervals from resampling sources. Table A35 gives the per-family AUCs. No confidence interval is obtained by averaging interval endpoints.

Table A34: ICP normalization variants on eight families per domain. Code-mean is the ICP-MIA baseline in Table 1. Intervals are saved 95% paired source-bootstrap AUC intervals.

Domain	G	Variant	AUC	95% CI	TPR_5	TPR_1
Enron	Same	Paper-sum	0.7180	[0.6935, 0.7423]	0.2438	0.1010
Enron	Same	Code-mean	0.7786	[0.7582, 0.7982]	0.3312	0.1035
Enron	Cross	Paper-sum	0.6416	[0.6152, 0.6677]	0.1581	0.0517
Enron	Cross	Code-mean	0.6909	[0.6678, 0.7130]	0.2096	0.0788
BillSum	Same	Paper-sum	0.7304	[0.7072, 0.7541]	0.2004	0.0530
BillSum	Same	Code-mean	0.8015	[0.7838, 0.8185]	0.3945	0.2090
BillSum	Cross	Paper-sum	0.6431	[0.6170, 0.6695]	0.1082	0.0281
BillSum	Cross	Code-mean	0.6911	[0.6699, 0.7117]	0.2243	0.0798

Table A35: Per-family ICP AUCs. Same/cross G average three/six generator pairs; all families have complete nine-pair coverage.

		Paper-sum		Code-mean	
Domain	Family	Same G	Cross G	Same G	Cross G
Enron	DeepSeek	0.7314	0.6533	0.8027	0.7073
Enron	Qwen3	0.7063	0.6321	0.7868	0.7002
Enron	SmolLM2	0.7093	0.6362	0.7770	0.6931
Enron	OLMo2	0.7097	0.6303	0.7750	0.6835
Enron	RedPajama	0.7372	0.6539	0.7843	0.6952
Enron	OpenLLaMA	0.7278	0.6485	0.8075	0.7150
Enron	Falcon	0.7228	0.6483	0.7640	0.6791
Enron	Pythia	0.6993	0.6298	0.7316	0.6537
BillSum	DeepSeek	0.7332	0.6487	0.8159	0.7077
BillSum	Qwen3	0.7207	0.6342	0.8234	0.7104
BillSum	SmolLM2	0.7212	0.6396	0.8044	0.6977
BillSum	OLMo2	0.7220	0.6362	0.7992	0.6881
BillSum	RedPajama	0.7415	0.6468	0.7985	0.6793
BillSum	OpenLLaMA	0.7394	0.6506	0.8289	0.7172
BillSum	Falcon	0.7424	0.6513	0.7829	0.6760
BillSum	Pythia	0.7229	0.6371	0.7585	0.6520