LOCALIZING VERBATIM MEMORIZED SPANS IN LANGUAGE MODEL OUTPUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) can reproduce training data verbatim, raising copyright and privacy concerns. To detect such reproduction, existing methods typically frame the problem as membership inference, classifying entire sequences as members or non-members. However, a long generated sequence may contain only a short memorized passage, for example, roughly 70 verbatim tokens within an up to 512-token output. Sequence-level aggregation can therefore dilute the memorization signal. While token-level scoring methods exist, they commonly flag tokens independently. Our key insight is that verbatim memorization should be treated as a structured span-localization problem rather than a sequence-level or independent-token detection problem. Unlike membership inference, which asks whether a candidate text was used in training, our task asks where verbatim matches occur within generated text. We make three contributions. First, we develop a benchmark for predicting the boundaries of verbatim spans and demonstrate the benefit of contiguity-aware extraction: on Hubble models, applying it to InfoRMIA improves mean intersection over union (IoU) from 0.326 to 0.683. Second, we introduce Token Span Localization (TSL), which converts token-level scores into contiguous span predictions using two scoring functions, Final Loss (FL) and Sample Loss Difference (SLD). Third, we provide a theoretical analysis showing that these scores offer complementary evidence: FL captures how likely the final model is to reproduce a complete span, while SLD measures how much its likelihood has increased since a reference checkpoint. On GPT-2 XL, TSL achieves 0.851 mean IoU (with SLD) and 0.984 detection AUC (with FL), a gain of 0.122 over the strongest sequence-level baseline (0.862). Across eight Hubble configurations, TSL obtains 0.986–0.993 AUC, an average gain of 0.207 over InfoRMIA (0.672–0.850), compared with 0.601–0.796 for the strongest conventional sequence-level baseline in each configuration. TSL also achieves the highest detection AUC among the evaluated methods on 12 of 14 open-weight models tested against an external corpus. Together, these results establish TSL as a more accurate and fine-grained approach to auditing verbatim memorization in LLM outputs.

1 INTRODUCTION

Large language models (LLMs) reproduce training data verbatim during generation, even when not explicitly prompted to do so (Carlini et al., 2021; 2023; Nasr et al., 2023). Such reproduction raises copyright and privacy concerns: a deployed model may emit copyrighted text or proprietary data without any adversarial prompting (Henderson et al., 2023). As models scale and train on increasingly large corpora, understanding *where* within a model's output such reproduction occurs becomes essential for responsible deployment.

Existing approaches to detecting memorized content typically frame the problem as membership inference: given a text sequence, classify it as a training member or non-member (Yeom et al., 2018; Shi et al., 2024; Zhang et al., 2025). These methods aggregate per-token signals into a single sequence-level statistic, discarding positional information. However, memorized content is sparse within generated text: we find that "memorized" up to 512-token generations contain only $\sim$70-token verbatim spans on average. Sequence-level aggregation dilutes the signal from short memorized passages across long outputs. This loss of granularity can both hide leakage and misdirect auditing. Tao & Shokri (2026) show that non-private tokens can drive high sequence-level membership scores,

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.

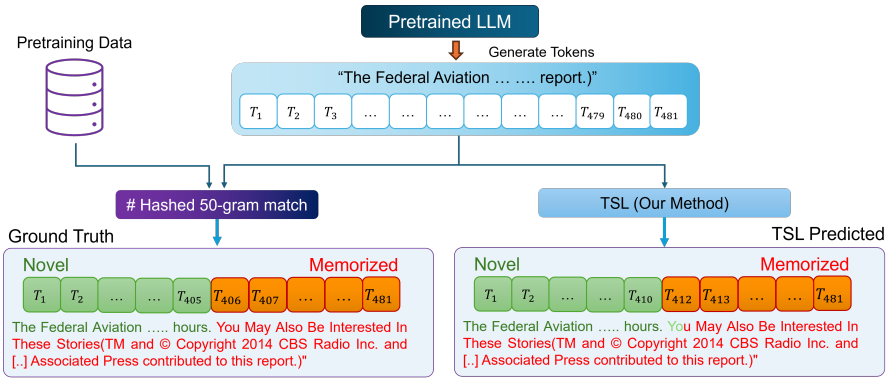


Figure 1: TSL localizes verbatim memorization spans in LLM-generated text. Contiguity-aware span extraction predicts the boundaries of verbatim-reproduced training content.

while averaging can obscure strong signals on private tokens. Three practical considerations motivate evaluating boundary accuracy. First, whole-sequence tests can miss partial reproduction: Dang & Mohaisen (2025) report subsequences that satisfy their multi-prefix memorization criterion even when the complete sequence does not. Second, attribution requires locating the overlap: Data Portraits (Marone & Van Durme, 2023) highlights training-overlap spans in generated code as a first step toward attribution. Third, targeted intervention requires separating affected text from its surroundings: in selective unlearning, Wang et al. (2025) find that forgetting sensitive spans better preserves generation quality than forgetting entire sequences.

To address this gap, we define *verbatim span localization*: given a sequence generated *unconditionally* by the model (i.e., prompted only with the beginning-of-sequence token), predict the start and end indices of a contiguous span that appears verbatim in the reference corpus, or abstain if none is present. The reference is the training corpus when available. Success is measured by boundary agreement (IoU) against corpus-derived spans, penalizing both missed portions of a reproduced passage and inclusion of surrounding text.

Our key insight is that verbatim memorization should be treated as a structured span-localization problem rather than a sequence-level or independent-token detection problem. Exploiting the contiguous structure of verbatim reproduction is what separates effective localization from naïve token-level flagging, we show that applying contiguity-aware span extraction to InfoRMIA's (Tao & Shokri, 2026) token-level scores improves mean IoU from 0.33 to 0.68, more than doubling mean IoU without changing the underlying scoring function.

In this paper, we propose Token Span Localization (TSL), a framework that combines per-token loss-based scoring with contiguity-aware span extraction. TSL thresholds per-token scores, retains only contiguous runs of flagged tokens, filters runs below a minimum length, and returns the longest surviving run as the predicted span. We employ two scoring functions: *Final Loss* (FL), which captures how likely the model is to reproduce a span, and *Sample Loss Difference* (SLD), which measures how much training has increased that likelihood. We formalize the connection between these scoring functions and verbatim reproduction probability (Proposition 4.2), showing that FL is the negative log-probability of verbatim reproduction and that SLD measures the training-induced amplification. These results are optimizer-independent and hold for any autoregressive model.

We evaluate TSL across three scenarios of increasing realism: (1) **Full control**: GPT-2 XL trained from scratch on OpenWebText (Gokaslan & Cohen, 2019); (2) **Open model and data**: Hubble 1B/8B models trained on 100B/500B tokens from DCLM (Li et al., 2024) (Wei et al., 2026); (3) **Open weights only**: 14 models evaluated against the external corpus DCLM-100B (Li et al., 2024). On GPT-2 XL, TSL achieves 0.851 mean IoU (TSL $_{\text{SLD}}$) and 0.984 detection AUC (TSL $_{\text{FL}}$), exceeding the strongest baseline (0.862) by 0.122. On Hubble models, TSL achieves 0.648–0.783 IoU and mean detection AUC of 0.989 versus 0.783 for the strongest baseline across eight configurations. For open-weight models, TSL achieves the highest detection AUC on 12 of 14 models. In summary:

- We develop a benchmark for verbatim span boundary prediction, evaluated across three scenarios, and show that contiguity-aware span extraction improves InfoRMIA's mean IoU from 0.33 to 0.68 using the same token-level scores.

- We propose TSL, a metric-agnostic span localization framework that exploits the contiguous structure of verbatim memorization. TSL achieves 0.851 mean IoU and 0.984 detection AUC on GPT-2 XL, maintains strong localization on Hubble models (0.648–0.783 IoU), and achieves the highest detection AUC on 12 of 14 open-weight models.
- We employ two loss-based scoring functions, FL and SLD, and provide a theoretical analysis showing that they offer complementary evidence: FL captures reproduction likelihood while SLD measures training-induced amplification. The theoretical guarantees are optimizer-independent.

2 Related Work

Verbatim memorization in LLMs. Carlini et al. (2021) demonstrated that language models emit training data verbatim from GPT-2. Carlini et al. (2023) formalized k-*extractable memorization* and showed that memorization scales with model size and data duplication. Nasr et al. (2023) extended extraction to production-scale models, and Hayes et al. (2025) introduced a probabilistic extraction framework with false-positive calibration. Huang et al. (2024) showed that non-trivial repetition is necessary for verbatim memorization, while Liu et al. (2025) showed that LLMs can verbatim complete sequences even after all matching n-grams have been removed from training data. Separately, *counterfactual memorization* (Feldman, 2020; Zhang et al., 2023) quantifies memorization via leave-one-out retraining but is prohibitive for LLMs. Rather than studying *whether or how much* models memorize, we detect and localize *where* verbatim reproduction occurs within model outputs.

Sequence-level pretraining data detection. Pretraining data detection can be viewed as membership inference (Shokri et al., 2017; Yeom et al., 2018; Maini et al., 2024), but poses distinct challenges for LLMs: pretraining corpora are often nonpublic, few-epoch training weakens memorization signals, and shadow model approaches are infeasible at scale (Shi et al., 2024; Duan et al., 2024). Shi et al. (2024) proposed Min-K% and the WikiMIA benchmark. Min-K%++ (Zhang et al., 2025) improved calibration and evaluated an "online detection" scenario using non-overlapping 32-token chunks of mixed member/non-member text. RMIA (Zarifzadeh et al., 2023) uses likelihood ratios against reference populations. These methods evaluate sequence- or chunk-level membership predictions rather than recovery of corpus-verified span boundaries.

Token-level scoring. InfoRMIA (Tao & Shokri, 2026) assigns per-token membership scores, providing finer granularity than sequence-level methods. However, InfoRMIA treats tokens independently and does not formulate a span boundary prediction task, exploit contiguity, or report localization metrics. Applying contiguity-aware span extraction to InfoRMIA scores improves mean IoU from 0.33 to 0.68, showing that the extraction procedure—not the score—drives localization performance.

Localization in model parameters. A separate line of work localizes memorization within the model's *weights* (Stoehr et al., 2024; Chang et al., 2024). Our use of "localization" is distinct: we localize memorization in the model's *output*, identifying which generated tokens are verbatim reproductions of training data.

3 Problem Formulation

Let p_θ be an autoregressive language model and let S denote the reference corpus available to the evaluator. Depending on the setting, S may be the complete training corpus, a known subset of it, or an external corpus used as an approximation. An *index of* S is the searchable data structure used to perform exact token-sequence lookup. Consequently, a match certifies overlap with S, and certifies overlap with the model's training data only when S is known to be part of that data. We sample a sequence $\mathbf{X} = (X_1, \ldots, X_N)$ from p_θ via unconditional generation, i.e., conditioned only on the beginning-of-sequence token.

Definition 3.1 (Verbatim span localization)**.** A contiguous subsequence $X_{s:e} = (X_s, \ldots, X_e)$ is a *verbatim hit* if $X_{s:e}$ occurs as a contiguous substring in S and $e - s + 1 \geq \gamma$, where γ is a minimum match length. The *verbatim span localization task* is: given a generated sequence $\mathbf{X}$, predict the start and end indices (s, e) of the verbatim hit, or abstain if no hit is present.

This task differs from related problems in two ways. First, unlike membership inference (Shokri et al., 2017; Shi et al., 2024; Zhang et al., 2025), our input is model-generated text and the output is a

span boundary prediction, not a binary member/non-member decision. Second, unlike discoverable extraction (Carlini et al., 2023; Nasr et al., 2023), generation is unconditional: the model reproduces training content spontaneously, without being prompted with known training prefixes.

We identify ground-truth verbatim hits by exact γ-gram matching against S, with $\gamma = 50$ throughout. For each generated sequence, we slide a γ-token window and query an index of S. A sequence is labeled positive if any window returns a match, and the longest matched region defines the ground-truth boundaries (s^*, e^*) (Appendix D.1).

For localization quality, we report intersection over union (IoU) defined:

$$\mathrm{IoU}([s^\star, e^\star], [\hat{s}, \hat{e}]) = \frac{|[s^\star, e^\star] \cap [\hat{s}, \hat{e}]|}{|[s^\star, e^\star] \cup [\hat{s}, \hat{e}]|}, \tag{1}$$

here both arguments are closed intervals of token positions and (s^*, e^*) are the ground-truth boundaries and $(\hat{s}, \hat{e})$ are the predicted boundaries. We report the mean IoU over all positive sequences: a positive for which TSL returns no span, or returns a span that does not overlap the ground truth, contributes $\mathrm{IoU} = 0$. Appendix D.1 gives the procedure for resolving matches to a single ground-truth span. For detection we report AUC. For localization we additionally report the overlap rate (fraction of predicted spans that share at least one token with the ground-truth span) and the Pearson correlation between predicted and ground-truth boundary positions, both computed over positives for which a span is predicted.

4 Token Span Localization

4.1 Scoring functions

For a candidate interval $[s, e]$ within a generated sequence $\mathbf{X} = (X_1, \ldots, X_N)$, define

$$x := X_{<s}, \qquad y := X_{s:e} = (X_s, \ldots, X_e), \qquad m := e - s + 1. \tag{2}$$

Thus x is the generated prefix before the candidate, y is the candidate span itself, and m is its length. For an indexed verbatim candidate we have $m \geq \gamma$. At checkpoint t, define the total span loss $L_t(y \mid x) := -\sum_{j=1}^{m} \log p_{\theta_t}(y_j \mid x, y_{<j})$, with natural logarithms and final checkpoint T. We assume positive token probabilities. Reproduction probabilities below refer to raw ancestral sampling from p_θ; modified decoders require scores computed from their own distributions (Remark A.4).

Definition 4.1 (Final Loss and Sample Loss Difference)**.** Final Loss (FL) measures the absolute likelihood assigned by the final model, whereas Sample Loss Difference (SLD) measures the change relative to a reference checkpoint t_r:

$$FL(y \mid x) := L_T(y \mid x) = -\log p_{\theta_T}(y \mid x), \tag{3}$$

$$SLD_{t_r}(y \mid x) := L_{t_r}(y \mid x) - L_T(y \mid x) = \log \frac{p_{\theta_T}(y \mid x)}{p_{\theta_{t_r}}(y \mid x)}. \tag{4}$$

FL asks whether the final model is likely to reproduce the span; SLD asks how much more likely the span became during training. Both are total span scores. We write FL_t and $SLD_t^{t_r}$ for per-token scores and $\overline{\mathrm{FL}}$, $\overline{\mathrm{SLD}}_{t_r}$ for mean per-token versions. Both span scores decompose as sums of per-token counterparts: $FL(y \mid x) = \sum_{j=1}^{m} FL_{s+j-1}$ and $\mathrm{SLD}_{t_r}(y \mid x) = \sum_{j=1}^{m} SLD_{s+j-1}^{t_r}$, by the autoregressive factorization. This is why contiguous runs of per-token scores, rather than scattered favorable positions, aggregate into span-level reproduction guarantees.

Proposition 4.2 (Reproduction certificate)**.** *For an indexed span y of length $m \geq \gamma$ and any desired reproduction level $\tau \in (0, 1)$:*

$$FL(y \mid x) \leq -\log \tau \iff p_{\theta_T}(y \mid x) \geq \tau \iff \mathrm{SLD}_{t_r}(y \mid x) \geq \log \frac{\tau}{p_{\theta_{t_r}}(y \mid x)}. \tag{5}$$

The probability that the model reproduces this indexed span after x is therefore at least τ. The formal event definition, its relation to a hit anywhere in a generation, and the proof are in Appendix A.1.

This certificate starts with a span already verified by corpus lookup. It does not infer corpus membership, recover unknown boundaries, or attribute the reproduction to a particular training record. The causal analysis below addresses attribution under additional assumptions; the span extraction algorithm addresses boundary prediction.

4.2 CAUSAL ATTRIBUTION

Proposition 4.2 certifies when a model is likely to reproduce a span, but does not say that one record caused that behavior: duplicates, related examples, and general language-modeling ability may all contribute (Huang et al., 2024). The relevant counterfactual question is whether the model would still reproduce the span if record z_i had been omitted from training. Following Feldman (2020); Zhang et al. (2023), let I_i and O_i be the expected final mean per-token losses when record z_i is included and omitted, and let R_i be the expected reference-checkpoint mean loss in the included run. All expectations use the same training-randomness convention and are finite. Since $\mathbb{E}[\overline{\mathrm{SLD}}_{t_r}(z_i)] = R_i - I_i$, we obtain

$$\mathrm{mem}_{\mathrm{loss}}(z_i) := O_i - I_i = \mathbb{E}[\overline{\mathrm{SLD}}_{t_r}(z_i)] - (R_i - O_i). \tag{6}$$

The first term is the learning observed after the reference checkpoint; the second removes improvement compatible with omitting z_i. High SLD is evidence of learning during training, but is not by itself evidence of record-level causation. We formalize causation through Probability of Necessity (PN) and Probability of Necessity and Sufficiency (PNS).

Definition 4.3 (Causal verbatim outcomes)**.** Let $H_i(a)$ indicate exact reproduction of y after x when z_i is included ($a=1$) or omitted ($a=0$), with all other aspects matched. Here probabilities are over coupled decoding randomness, conditional on a realized pair of included and omitted training runs. Let $p_a := \Pr\{H_i(a) = 1\}$ for $a \in \{0, 1\}$, with $p_1 > 0$. Define

$$\mathrm{PN}_i := \Pr\{H_i(0) = 0 \mid H_i(1) = 1\}, \qquad \mathrm{PNS}_i := \Pr\{H_i(1) = 1, H_i(0) = 0\}. \tag{7}$$

To bound PN and PNS, we need information about what can be learned when the source record is absent from training. If we had K records similar to the target record z_i, but excluded from training, then the checkpoint improvements of their corresponding spans would show how much target-like content can improve without its own record. We can use these matched excluded controls to calibrate an upper mean-improvement level $\bar{b}_\delta$. With this we can state

Theorem 4.4 (Confidence-calibrated causal certificate)**.** *Let $\bar{S}_i := \overline{\mathrm{SLD}}_{t_r}(y \mid x)$ and $\bar{F}_i := \overline{\mathrm{FL}}(y \mid x)$ be the target span's mean per-token SLD and FL, and let $\bar{d}_{i,\delta} := [\bar{S}_i - \bar{b}_\delta]_+$, where $[u]_+ := \max\{u, 0\}$. Under the matched-control assumptions in Appendix A.5, both bounds hold for one pre-specified target with calibration coverage at least $1 - \delta$:*

$$\mathrm{PN}_i \geq 1 - e^{-m\bar{d}_{i,\delta}}, \qquad \mathrm{PNS}_i \geq e^{-m\bar{F}_i}\left(1 - e^{-m\bar{d}_{i,\delta}}\right). \tag{8}$$

The PN/PNS certificate requires matched excluded controls satisfying the stated exchangeability or conservative-domination assumption; similarity alone does not establish it. Coverage is marginal over the target/control calibration law, conditional on the specified matching information. Reusing a global $\bar{b}_\delta$ does not by itself give simultaneous coverage across targets. Our validation-chosen localization threshold is not this causal calibration bound, and the experiments do not validate PN/PNS coverage. Full assumptions and multi-target corrections are in Appendix A.

4.3 SPAN EXTRACTION ALGORITHM

TSL first flags positions with sufficiently low FL or sufficiently high SLD. A left-to-right scan groups adjacent flagged positions into maximal runs, closing each run before the next unflagged token. It discards runs shorter than $L_{\min}$ and returns the longest remaining run, or abstains if none survives. Endpoints are inclusive, so a run $[s, e]$ has length $e - s + 1$; ties can be resolved by the smallest start index. This scan takes $O(N)$ time once token scores are available. It uses contiguity to turn token scores into a boundary prediction; it does not require corpus lookup at prediction time. At position t, the per-token scores are

$$FL_t := -\log p_{\theta_T}(X_t \mid X_{<t}), \tag{9}$$

$$SLD_t^{t_r} := -\log p_{\theta_{t_r}}(X_t \mid X_{<t}) + \log p_{\theta_T}(X_t \mid X_{<t}). \tag{10}$$

Algorithm 1 Token Span Localization (TSL)

Require: Per-token scores $(s_1, \ldots, s_N)$, threshold ζ, minimum span length $L_{\min}$, direction $d \in \{\text{low}, \text{high}\}$
Ensure: Predicted span $(\hat{s}, \hat{e})$ or $\varnothing$
if $d = \text{low}$ **then**
  flagged $\leftarrow \{t : s_t \leq \zeta\}$ {FL: low loss}
else
  flagged $\leftarrow \{t : s_t \geq \zeta\}$ {SLD: high amplification}
end if
Identify all maximal contiguous runs in flagged
Discard runs with length $< L_{\min}$
if no runs remain **then**
  return $\varnothing$
end if
return the longest surviving run $(\hat{s}, \hat{e})$

For a returned run of length m, the token-to-span aggregation gives

$$FL(y \mid x) \leq m\zeta \implies p_{\theta_T}(y \mid x) \geq e^{-m\zeta} \tag{11}$$

for FL. For SLD, summing the m inequalities $SLD_t^{t_r} \geq \zeta$ gives

$$SLD_{t_r}(y \mid x) \geq m\zeta \implies \frac{p_{\theta_T}(y \mid x)}{p_{\theta_{t_r}}(y \mid x)} \geq e^{m\zeta}. \tag{12}$$

Corpus lookup, not score thresholding alone, determines whether the run is an indexed verbatim hit. Appendix A.3 proves the aggregation bounds and shows that the algorithm solves the longest-run problem induced by the tokenwise threshold. The threshold ζ, minimum length $L_{\min}$, and boundary accuracy remain empirical choices. The procedure applies to any token-level score. We choose ζ and $L_{\min}$ on a held-out validation set by maximizing detection AUC, then freeze them for test evaluation (Appendix E.2).

5 Experiments

We evaluate across three scenarios with decreasing levels of data access. **Scenario 1, Full control:** The evaluator trains the model and controls the training data. The reference corpus S is the complete training set. Both FL and SLD are computable. **Scenario 2, Open model and data:** Model weights, training data, and intermediate checkpoints are publicly available. The reference corpus S is the known training set. Both FL and SLD are computable (using an early checkpoint as the reference). **Scenario 3, Open weights only:** Only model weights are public. No training data or checkpoints are available, so only FL (requiring a single forward pass) is applicable. The reference corpus S is an external corpus, and whose relationship to the unknown training data is unverified. In Scenarios 1 and 2, a match is direct evidence that the model reproduced training content. In Scenario 3, we measure *external-corpus overlap*: a match indicates that a generated span appears verbatim in S, but does not establish that it was memorized from training data, and a non-match does not rule out memorization from sources outside S. Because both errors are possible, we make no claim about the direction of bias relative to true training-data memorization. We denote TSL_{SLD} when using SLD as the scoring function and TSL_{FL} when using FL.

5.1 Experimental setup

Models and generation protocol. Scenario 1 uses GPT-2 XL trained from scratch on OpenWebText (Gokaslan & Cohen, 2019). Scenario 2 uses Hubble 1B and 8B models (Wei et al., 2026) trained on 100B and 500B tokens from DCLM (Li et al., 2024), in both standard and perturbed variants (eight configurations). Scenario 3 uses 14 open-weight models from Pythia (Biderman et al., 2023), DCLM (Li et al., 2024), Falcon (Almazrouei et al., 2023), Llama-3 (Grattafiori et al., 2024), Gemma-2 (Gemma Team, 2024), and Nemotron-H (NVIDIA, 2025) (410M to 12B parameters), spanning models with documented DCLM overlap and models with closed training corpora. We generate

sequences of up to 512 tokens unconditionally (BOS only) with top-k and nucleus sampling (k=40, p=0.96) following Carlini et al. (2021), at T=1.0 for Scenarios 1–2 and T=0.7 for Scenario 3. Scenarios 1–2 use ≈50K validation and ≈100K disjoint test sequences per model; Scenario 3 uses up to 50K per model, split 50/50. Per-model counts and hit rates are in Appendix Table 4.

Index construction. We construct the ground-truth index by sliding a 50-token window with stride 1 over each document in S, hashing each window (xxh3_128 for Scenarios 1–2, xxh3_64 for Scenario 3), and storing mappings from hash to (document_id, offset) in LMDB. This enables $O(1)$ lookup per window. The fraction of generated sequences containing a verbatim hit ranges from 0.03% to 2.20% across settings (Appendix D.4).

Baselines. We compare against sequence perplexity (PPL); likelihood-based heuristics from Carlini et al. (2021) (Zlib, Lowercase, Smaller Ref); Min-K% (Shi et al., 2024); Min-K%++ (Zhang et al., 2025); Neighbor (Mattern et al., 2023); and InfoRMIA (Tao & Shokri, 2026). These methods produce a single sequence-level score without predicting span boundaries.

Hyperparameter tuning. The threshold ζ and minimum span length $L_{\min}$ are grid searched on the validation set to maximize detection AUC. Hyperparameters are frozen for test evaluation.

5.2 Scenario 1: Full control (GPT-2 XL)

Both SLD (using the randomly initialized checkpoint as reference) and FL are evaluated. Table 1 reports results. Both achieve strong localization: SLD reaches mean IoU of 0.851 and FL reaches 0.801. More than 96% of predicted spans overlap the ground truth, and boundary Pearson correlations exceed 0.95. For detection, TSL_{FL} achieves AUC 0.984, outperforming the strongest baseline (PPL, 0.862) by 0.122. Baseline AUCs span 0.722–0.862; InfoRMIA reaches 0.838. Permuting each sequence's per-token scores while holding their values fixed collapses AUC from 0.98 to ≈ 0.51 (Appendix D.5), isolating contiguity as the property TSL exploits.

5.3 Scenario 2: Open model and data (Hubble)

Both FL and SLD (using an early checkpoint as reference) are evaluated across all eight configurations. Table 1 reports results. Mean IoU ranges from 0.648 to 0.783, with Pearson correlations consistently above 0.83. Perturbed training data improves localization in seven of the eight matched standard/perturbed comparisons (two scoring functions × two model sizes × two token budgets). The largest gain is for the 1B model at 500B tokens, where FL reaches mean IoU of 0.783 (vs 0.716 standard) with boundary Pearson correlations exceeding 0.98; the only exception is FL on the 8B model at 500B tokens (0.714 vs 0.724), and the size of the gain does not follow a consistent pattern with model scale. FL yields a higher overlap rate than SLD in six of eight configurations (e.g., 99.4% vs 98.3% for 500B perturbed 1B). For detection, TSL_{FL} achieves AUC >0.98 on all configurations, while conventional sequence-level baselines remain weaker (0.403–0.796 across all configurations).

5.4 Scenario 3: Open weights only (14 models)

Only FL is applicable (no reference checkpoints). Hit rates range from 0.03% to 1.80% (Table 4 in Appendix). For models with closed training corpora (Llama-3, Gemma-2), matches indicate DCLM-100B overlap, but non-matches do not rule out memorization from undisclosed sources.

Localization. Table 2 reports results. Localization is strongest on Falcon-RW-1B (mean IoU 0.715) and TRI-ML/DCLM-1B (0.671), followed by apple/DCLM-7B (0.625). Overlap rates exceed 86% for these models, reaching 96.9% on Falcon-RW-1B. Boundary Pearson correlations reach 0.90 (start) and 0.90 (end) on TRI-ML/DCLM-1B; on apple/DCLM-7B and Falcon-RW-1B one boundary is well localized (0.91 end and 0.88 start, respectively) while the other is weaker (0.53 and 0.60). IoU is lower for Llama-3 and Gemma-2, whose training corpora overlap DCLM-100B only to an unknown extent; since ground-truth spans are defined by the external index, these values measure agreement with S rather than with the models' true memorized content, and may differ from it in either direction.

Detection. TSL_{FL} achieves the highest AUC on 12 of 14 models. Strongest AUCs are apple/DCLM-7B (0.992), Falcon-RW-1B (0.971), and TRI-ML/DCLM-1B (0.955); the apple/DCLM-7B value rests on very few test positives (14 hits in total, Table 4), and on a larger disjoint test set with 31 hits it is 0.983 (Table 10). Gemma-2 models are exceptions where sequence-level loss baselines perform

Table 1: Results for token-span localization performance and detection (AUC) on controlled models. Score direction is fixed a priori from each method's definition and is not selected per configuration, so a method whose signal is inverted in a given setting scores below 0.5. †Neighbor is computationally expensive to scale to larger datasets.

			GPT-2 XL	**Hubble (100B tokens)**				**Hubble (500B tokens)**			
			OpenWebText	Standard		Perturbed		Standard		Perturbed	
Task	**Proxy**	**Metric / Method**	XL	1B	8B	1B	8B	1B	8B	1B	8B
Localization	SLD	IoU ↑	**0.851**	0.679	0.669	0.740	0.692	0.743	0.648	0.751	0.660
		Pearson (start) ↑	0.954	0.878	0.851	0.889	0.833	0.949	0.841	**0.975**	0.860
		Pearson (end) ↑	0.952	0.849	0.860	0.867	0.844	0.919	0.858	**0.967**	0.859
		Overlap (%) ↑	96.2	93.3	93.2	95.2	92.3	97.4	93.0	**98.3**	93.1
	FL	IoU ↑	**0.801**	0.738	0.663	0.776	0.725	0.716	0.724	0.783	0.714
		Pearson (start) ↑	0.968	0.941	0.831	0.944	0.873	0.914	0.914	**0.988**	0.923
		Pearson (end) ↑	0.966	0.911	0.838	0.925	0.880	0.878	0.928	**0.985**	0.926
		Overlap (%) ↑	97.5	96.0	92.8	96.8	94.1	95.5	95.9	**99.4**	95.6
Detection (AUC)		TSL_{FL}	**0.984**	**0.990**	**0.989**	**0.993**	**0.988**	**0.992**	**0.987**	**0.989**	**0.986**
		TSL_{SLD}	0.982	0.988	0.979	0.989	0.976	0.990	0.971	0.988	0.982
		PPL	0.862	0.561	0.582	0.444	0.516	0.503	0.555	0.461	0.518
		Zlib	0.861	0.528	0.543	0.413	0.472	0.461	0.521	0.432	0.479
		Smaller Ref	0.855	0.493	0.568	0.417	0.570	0.436	0.569	0.403	0.653
		InfoRMIA	0.838	0.754	0.800	0.785	0.843	0.712	0.844	0.672	0.850
		Lowercase	0.809	0.608	0.595	0.471	0.514	0.573	0.580	0.549	0.531
		Neighbor†	0.799	—	—	—	—	—	—	—	—
		Min-k%	0.768	0.680	0.738	0.731	0.796	0.613	0.738	0.601	0.752
		Min-k%++	0.722	0.591	0.621	0.663	0.693	0.536	0.675	0.496	0.671

Table 2: Localization and membership-inference performance across 14 open-weight models, reported on the test half of a deterministic 50/50 validation/test split. Mean IoU is computed over test-split hits. TSL uses hyperparameters $(\zeta, L_{\min})$ tuned on the validation split only. Sequence-level baseline AUCs (Mean loss, Min loss, Min-$K\%$, zlib ratio) are computed per generation.

	Localization				**Detection (AUC)**				
Model	IoU	Pearson (start)	Pearson (end)	Overlap (%)	Mean	Min	Min-$K\%$	Zlib	TSL
TRI-ML/DCLM-1B	0.671	**0.901**	0.895	86.4	0.698	0.879	0.906	0.631	**0.955**
apple/DCLM-7B	0.625	0.532	**0.913**	87.5	0.631	0.831	0.847	0.810	**0.992**
Llama-3.2-1B	0.193	-0.202	0.130	86.7	0.838	0.572	0.522	0.833	**0.917**
Llama-3.1-8B	0.304	0.597	0.595	82.4	0.812	0.538	0.508	0.821	**0.945**
Gemma-2-2B	0.072	-0.029	-0.185	48.4	0.881	**0.910**	0.906	0.870	0.880
Gemma-2-9B	0.058	-0.122	-0.277	43.1	**0.843**	0.815	0.837	0.776	0.826
Falcon-RW-1B	**0.715**	0.881	0.602	**96.9**	0.945	0.647	0.859	0.646	**0.971**
Falcon-7B	0.418	0.615	0.241	84.5	0.899	0.683	0.701	0.748	**0.915**
Nemotron-H-8B	0.606	**0.901**	0.664	88.6	0.748	0.524	0.594	0.778	**0.879**
Pythia-410M	0.486	0.775	0.523	88.2	0.847	0.606	0.819	0.547	**0.948**
Pythia-1.4B	0.504	0.781	0.509	87.5	0.853	0.671	0.802	0.591	**0.945**
Pythia-2.8B	0.426	0.595	0.536	78.7	0.913	0.833	0.920	0.562	**0.938**
Pythia-6.9B	0.579	0.808	0.612	87.4	0.804	0.508	0.711	0.525	**0.905**
Pythia-12B	0.444	0.715	0.382	84.8	0.790	0.555	0.677	0.612	**0.896**

slightly better. Sequence-level baselines vary widely across models (0.508–0.945), while TSL_{FL} remains above 0.82 across all models.

5.5 Analysis

Extraction, not the scoring function, drives localization. Applying our span extraction (Algorithm 1) to InfoRMIA's own token scores (Tao & Shokri, 2026) raises mean IoU from 0.326 to 0.683, consistently across all eight Hubble configurations (Table 3). Likewise, SLD and FL under independent per-token flagging reach only 0.267 and 0.272, rising to 0.698 and 0.730 with extraction. Every score gains 0.28–0.53 IoU from extraction, while the choice among scores shifts results by at

Table 3: Isolating the contribution of span extraction from the scoring function (mean IoU ↑). Each scoring function is shown twice: with independent per-token flagging, and with our span extraction applied to the same token scores. No predicted span counts as $\mathrm{IoU} = 0$.

	Hubble (100B tokens)				**Hubble (500B tokens)**				
	Standard		Perturbed		Standard		Perturbed		
Method	1B	8B	1B	8B	1B	8B	1B	8B	**Avg**
InfoRMIA (published)	0.291	0.308	0.388	0.388	0.304	0.312	0.316	0.299	0.326
+ extraction	0.663	0.646	0.693	0.664	0.725	0.648	0.734	0.687	0.683
SLD	0.253	0.257	0.336	0.323	0.249	0.230	0.244	0.241	0.267
+ extraction (TSL $_{\text{SLD}}$)	0.679	0.669	0.740	0.692	**0.743**	0.648	0.751	0.660	0.698
FL	0.254	0.262	0.337	0.336	0.249	0.236	0.252	0.247	0.272
+ extraction (TSL $_{\text{FL}}$)	**0.738**	0.663	**0.776**	**0.725**	0.716	**0.724**	**0.783**	**0.714**	**0.730**

most 0.083: memorized spans are contiguous by construction (Definition 3.1), so independent token scoring discards structure that extraction recovers.

FL and SLD provide complementary signals. FL requires only the final model (a single forward pass) and measures reproduction probability. SLD additionally requires a reference checkpoint and measures training-time amplification. In Scenarios 1 and 2, both achieve comparable localization (Table 1). In Scenario 3, FL achieves strong performance, making it the most broadly applicable.

6 Discussion

TSL detects verbatim (exact-match) reproduction of contiguous spans and localizes it precisely where the training corpus is fully indexed. Identifying the specific passage makes auditing more actionable: auditors can inspect the relevant content rather than reviewing an entire flagged output. Paraphrastic memorization requires separating matcher from detector: an embedding-based matcher redefines which generations count as memorized, supplying a new target rather than a method for it. The per-token proxies are not tied to surface form and plausibly transfer; the span extractor's contiguity criterion is inherited from exact matching, which paraphrase breaks. Replacing positional with semantic adjacency there is the open problem. The benefit of span structure also extends beyond our chosen scoring functions: applying contiguous extraction to InfoRMIA improves mean IoU from 0.326 to 0.683, demonstrating that span-aware extraction can substantially strengthen existing token-level scores. The current scoring functions require access to token probabilities, an access requirement shared by all established baselines (PPL, Min-K%, Min-K%++, and neighborhood comparison (Shi et al., 2024; Zhang et al., 2025; Mattern et al., 2023)). FL-based TSL needs only the final model, making it particularly practical for auditing open-weight models and potentially APIs exposing the required token log probabilities.

7 Conclusion

We formulated verbatim span localization as the task of predicting the boundaries of training-data spans reproduced in LLM-generated text, and showed that treating it as a structured span-localization problem rather than a sequence-level or independent-token detection problem substantially improves performance. Contiguity-aware extraction raises mean IoU for every scoring function we test, from $+0.36$ (InfoRMIA) to $+0.46$ (FL): the extraction procedure dominates the choice of scoring function, though it does not render it irrelevant. We proposed TSL, which combines per-token loss-based scoring with contiguity-aware span extraction. The connection between per-token loss and verbatim reproduction probability is formalized through results that hold for any autoregressive model. TSL achieves 0.851 mean IoU (TSL $_{\text{SLD}}$) and 0.984 detection AUC (TSL $_{\text{FL}}$) on GPT-2 XL, 0.648–0.783 IoU across eight Hubble configurations, and the highest detection AUC on 12 of 14 open-weight models. These results establish that loss-based span localization enables scalable, fine-grained auditing of verbatim memorization in LLM outputs.

AI USE STATEMENT

We used two large language models (Claude, Anthropic, GPT, OpenAI) as assistants during the revision of this paper. Its use fell into the following categories.

Tasks requiring disclosure. We used the model to help reorganize the presentation of our theoretical results, specifically, restructuring the SLD family around an explicit reference checkpoint t_r and consolidating redundant statements, and to check our existing proofs for internal consistency and notational errors. The theoretical claims, definitions, and proof strategies are our own; We used it to obtain feedback on experimental design: in particular, it identified that our token-order ablation as originally described conflated two distinct questions, and helped us specify the score-permutation variant reported in Appendix D.5. We used it to help interpret and verify results, including diagnosing an error in an earlier version of our AUC reporting convention. We did not use generative AI to generate synthetic datasets, to clean or reformat datasets, to assist with translation, or to implement the methods evaluated in this paper.

Tasks with recommended disclosure. We used the model to write diagnostic and analysis scripts used to verify reported statistics (hit rates, confidence intervals, and consistency checks between tables); to draft and edit portions of the manuscript text for clarity and concision; to suggest a structure for the appendix; and to propose candidate titles.

Verification. We have reviewed all AI-assisted work. All numerical values reported in this paper were computed by scripts we ran ourselves on our own data and were checked against the underlying output files by at least one author. All proofs were verified by the authors independently of the model's consistency checks. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Dual use. Our method locates verbatim reproductions of training data within model outputs. The same capability that supports auditing could in principle assist an adversary seeking to extract training content. Two features of the setting limit this concern. First, establishing ground truth requires a searchable index of the training corpus; an adversary who already holds the training data has no need to localize reproductions of it. Second, the scores themselves certify nothing about corpus membership, as we note in Appendix A.3, neither FL nor SLD thresholding proves that a span occurs in the training set. The method's practical value is therefore to a party who holds the corpus, which is the auditing party: a model developer, a data owner, or a regulator acting with the developer's cooperation.

Reproduced content. The qualitative examples in Appendix B.1 display spans reproduced verbatim by the models we study. These come from public web corpora (OpenWebText and DCLM) and were reviewed by the authors to exclude personal or sensitive information; the retained examples consist of news text and site boilerplate. We show them because the phenomenon under study cannot be illustrated otherwise, and we limit the number and length of displayed spans to what is needed for that purpose.

What our results do not establish. A negative result from our method is not evidence that a model has not memorized a given passage. Coverage is limited by what the reference index contains, by the minimum match length γ, and by the restriction to verbatim reproduction; paraphrastic and structural memorization are outside the scope of the framework entirely. We state this explicitly because the asymmetry matters for deployment: our method can support a claim that memorization is present, but should not be used to support a claim that a model is free of it.

Human subjects, data, and models. This work involves no human subjects and required no IRB review. We introduce no new dataset. All evaluated open-weight models are publicly released, and the GPT-2 XL model we train ourselves uses the public OpenWebText corpus.

Reproducibility Statement

Complete proofs of all theoretical claims, together with the assumptions each requires, are given in Appendix A: the FL/SLD probability identities and their threshold certificates in Appendix A.1, the correspondence between the span identities and our algorithm in Appendix A.3, the assumption-free counterfactual decomposition in Appendix A.4, and the causal bounds with their calibration conditions (A1)–(A4) stated explicitly in Appendix A.5. The localization procedure is given as pseudocode in Algorithm 1, and the detection-evaluation procedure, including how per-token scores are reduced to one score per sequence and how sequences with no predicted span are ranked, in Algorithm 3.

For the experiments, Table 4 is the authoritative record of sampling parameters, generation counts, and verbatim-hit counts for every model and configuration, and all sample sizes reported elsewhere derive from it. Appendix D.1 specifies how corpus matches are resolved to a single reference interval and how generations matching multiple source documents are handled; Section 3 states the IoU convention used in every reported result. Appendix C.2 states the score-direction convention applied uniformly to our method and all baselines. Appendix E.2 reports sensitivity to the span threshold ζ and documents the validation protocol by which ζ and $L_{\min}$ are selected and then frozen before test evaluation. Training details for the GPT-2 XL model we train ourselves are in Appendix C.3, and hardware, runtimes, and total compute for every stage are in Appendix C.4. All other evaluated models are publicly released checkpoints. The source code is included in supplementary material.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, 2023.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TatRHT_1cK.

Ting-Yun Chang, Jesse Thomason, and Robin Jia. Do localization methods actually localize memorized data in llms? a tale of two benchmarks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3190–3211, 2024.

Trung Cuong Dang and David Mohaisen. Memories retrieved from many paths: A multi-prefix framework for robust detection of training data leakage in large language models. *arXiv preprint arXiv:2511.20799*, 2025.

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? *arXiv preprint arXiv:2402.07841*, 2024.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd annual ACM SIGACT symposium on theory of computing*, pp. 954–959, 2020.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

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

Jamie Hayes, Marika Swanberg, Harsh Chaudhari, Itay Yona, Ilia Shumailov, Milad Nasr, Christopher A Choquette-Choo, Katherine Lee, and A Feder Cooper. Measuring memorization in language models via probabilistic extraction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9266–9291, 2025.

Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A Lemley, and Percy Liang. Foundation models and fair use. *Journal of Machine Learning Research*, 24(400):1–79, 2023.

Jing Huang, Diyi Yang, and Christopher Potts. Demystifying verbatim memorization in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10711–10732, 2024.

Andrej Karpathy. nanogpt. https://github.com/karpathy/nanoGPT, 2022.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

Ken Ziyu Liu, Christopher A Choquette-Choo, Matthew Jagielski, Peter Kairouz, Sanmi Koyejo, Percy Liang, and Nicolas Papernot. Language models may verbatim complete text they were not explicitly trained on. *arXiv preprint arXiv:2503.17514*, 2025.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset? *Advances in Neural Information Processing Systems*, 37:124069–124092, 2024.

Marc Marone and Benjamin Van Durme. Data portraits: Recording foundation model training data. *Advances in Neural Information Processing Systems*, 36:15121–15135, 2023.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 11330–11343, 2023.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

NVIDIA. Nemotron-h: A family of accurate and efficient hybrid mamba-transformer models. *arXiv preprint arXiv:2504.03624*, 2025.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zWqr3MQuNs.

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. IEEE, 2017.

Niklas Stoehr, Mitchell Gordon, Chiyuan Zhang, and Owen Lewis. Localizing paragraph memorization in language models. *arXiv preprint arXiv:2403.19851*, 2024.

Jiashu Tao and Reza Shokri. Information-theoretic membership inference for granular quantification of memorization. In *The Fourteenth International Conference on Learning Representations*, 2026.

Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 843–851, 2025.

Johnny Wei, Ameya Godbole, Mohammad Aflah Khan, Ryan Yixiang Wang, Xiaoyuan Zhu, James Flemings, Nitya Kashyap, Krishna P. Gummadi, Willie Neiswanger, and Robin Jia. Hubble: a model suite to advance the study of LLM memorization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZfdnZhOP0k.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282. IEEE, 2018.

Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. *arXiv preprint arXiv:2312.03262*, 2023.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=67o9UQgTD0.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-k%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZGkfoufDaU.

Appendix Table of Contents

A Proofs for the Verbatim-Span Theory **15**

A.1 Pointwise FL/SLD identities and proof of Proposition 4.2 15

A.2 Token decomposition of FL and SLD 16

A.3 TSL as a greedy token-threshold implementation 17

A.4 Counterfactual memorization and the exact SLD decomposition 18

A.5 Supporting causal bounds and proof of Theorem 4.4 18

A.5.1 Sharp paired-model bounds 20

A.5.2 Rank-calibrated single-run certificate 21

B Qualitative Examples and Algorithmic Details **23**

B.1 Qualitative Examples 23

B.2 Algorithms 24

C Experimental Setup and Compute **24**

C.1 Generation and Evaluation Protocol 24

C.2 Score Direction 27

C.3 GPT-2 XL Training Details 27

C.4 Computational Resources 28

D Ground-Truth Construction and Localization Diagnostics **29**

D.1 Ground-Truth Construction 29

D.2 ROC Curves 29

D.3 TSL Visualizations 30

D.4 Hit Rates on Hubble 30

D.5 Effect of Score Order on TSL 31

E Additional Counterfactual and Open-Weight Diagnostics **32**

E.1 Additional SLD^{+} Diagnostics 32

E.2 Additional TSL and Open-Weight Diagnostics 33

A Proofs for the Verbatim-Span Theory

A.1 Pointwise FL/SLD identities and proof of Proposition 4.2

For completeness, let S be the indexed corpus and let $\mathcal{C}_m(S)$ be the set of distinct length-m token strings that occur in it. For a generated sequence $\mathbf{X} = (X_1, \ldots, X_N)$, the formal indexed-hit event is

$$\mathcal{M}_\gamma(\mathbf{X}, S) := \mathbb{I}[\exists\, s, e :\ e - s + 1 \geq \gamma, X_{s:e} \in \mathcal{C}_{e-s+1}(S)]\,. \tag{13}$$

Here $\mathbb{I}[A]$ equals one when statement A is true and zero otherwise. Fix s, a prefix x, and a target span $y = (y_1, \ldots, y_m)$ of length $m := |y|$, with $1 \leq s \leq s + m - 1 \leq N$ and $\Pr\{X_{<s} = x\} > 0$. Probabilities in this subsection use raw ancestral sampling, with θ_T fixed. If a candidate was selected from an observed output, these statements concern a fresh continuation after its fixed prefix, not the probability of selecting that candidate again. Under the conditions of Proposition 4.2, the formal hit-probability conclusion is

$$\Pr\{\mathcal{M}_\gamma(\mathbf{X}, S) = 1 \mid X_{<s} = x, \theta_T\} \geq \tau. \tag{14}$$

For any fixed span y, prefix x, and reference checkpoint $t_r < T$,

$$p_{\theta_T}(y \mid x) = e^{-FL(y|x)} = p_{\theta_{t_r}}(y \mid x) e^{\mathrm{SLD}_{t_r}(y|x)}, \tag{15}$$

$$\mathrm{SLD}_{t_r}(y \mid x) = \log \frac{p_{\theta_T}(y \mid x)}{p_{\theta_{t_r}}(y \mid x)}. \tag{16}$$

Thus FL is the negative log final probability of the exact continuation, and SLD is its log probability ratio between the reference and final checkpoints. For comparison with the mean scores used empirically, define

$$\begin{aligned} \overline{\mathrm{FL}}(y \mid x) &:= -\frac{1}{m} \sum_{j=1}^{m} \log p_{\theta_T}(y_j \mid x, y_{<j}), \\ \overline{\mathrm{SLD}}_{t_r}(y \mid x) &:= \frac{1}{m} \sum_{j=1}^{m} \left[-\log p_{\theta_{t_r}}(y_j \mid x, y_{<j}) + \log p_{\theta_T}(y_j \mid x, y_{<j}) \right]. \end{aligned} \tag{17}$$

Summing these token-level definitions gives $FL(y \mid x) = m\overline{\mathrm{FL}}(y \mid x)$ and $\mathrm{SLD}_{t_r}(y \mid x) = m\overline{\mathrm{SLD}}_{t_r}(y \mid x)$. Their equivalent mean-score form is

$$p_{\theta_T}(y \mid x) = e^{-m\overline{\mathrm{FL}}(y|x)} = p_{\theta_{t_r}}(y \mid x) e^{m\overline{\mathrm{SLD}}_{t_r}(y|x)}. \tag{18}$$

Consequently, the reproduction certificate can equivalently be written as

$$\overline{\mathrm{FL}}(y \mid x) \leq -\frac{\log \tau}{m} \quad \Longleftrightarrow \quad \overline{\mathrm{SLD}}_{t_r}(y \mid x) \geq \frac{1}{m} \log \frac{\tau}{p_{\theta_{t_r}}(y \mid x)}. \tag{19}$$

Proof. For $y = (y_1, \ldots, y_m)$, the autoregressive chain rule gives

$$\begin{aligned} p_{\theta_T}(y \mid x) &= \prod_{j=1}^{m} p_{\theta_T}(y_j \mid x, y_{<j}) \\ &= \exp\left\{ \sum_{j=1}^{m} \log p_{\theta_T}(y_j \mid x, y_{<j}) \right\} \\ &= e^{-L_T(y|x)} = e^{-FL(y|x)}. \end{aligned} \tag{20}$$

The same identity at the reference checkpoint implies

$$p_{\theta_T}(y \mid x) = e^{-L_{t_r}(y|x)} e^{L_{t_r}(y|x) - L_T(y|x)} = p_{\theta_{t_r}}(y \mid x) e^{\mathrm{SLD}_{t_r}(y|x)}. \tag{21}$$

Together these equalities prove Equation 15. Taking the logarithm of the probability ratio proves Equation 16.

Fix $\tau \in (0, 1)$. Because the logarithm is strictly increasing, Equation 15 gives

$$FL(y \mid x) \leq -\log \tau \iff -FL(y \mid x) \geq \log \tau \tag{22}$$
$$\iff e^{-FL(y|x)} \geq \tau \tag{23}$$
$$\iff p_{\theta_T}(y \mid x) \geq \tau. \tag{24}$$

Likewise,

$$p_{\theta_T}(y \mid x) \geq \tau \iff p_{\theta_{t_r}}(y \mid x) e^{\mathrm{SLD}_{t_r}(y|x)} \geq \tau \tag{25}$$
$$\iff e^{\mathrm{SLD}_{t_r}(y|x)} \geq \frac{\tau}{p_{\theta_{t_r}}(y \mid x)} \tag{26}$$
$$\iff \mathrm{SLD}_{t_r}(y \mid x) \geq \log \frac{\tau}{p_{\theta_{t_r}}(y \mid x)}. \tag{27}$$

The division is valid because an ordinary softmax language model assigns strictly positive probability to every token and hence to every finite span. Dividing both sides by m proves Equation 19.

Finally, $y \in \mathcal{C}_m(S)$ and $m \geq \gamma$. Hence $\{X_{s:s+m-1} = y\}$ is a subset of $\{\mathcal{M}_\gamma(\mathbf{X}, S) = 1\}$. Monotonicity of probability under event inclusion gives

$$\Pr\{\mathcal{M}_\gamma(\mathbf{X}, S) = 1 \mid X_{<s} = x, \theta_T\} \geq \Pr\{X_{s:s+m-1} = y \mid X_{<s} = x, \theta_T\} \tag{28}$$
$$= p_{\theta_T}(y \mid x) \geq \tau, \tag{29}$$

which proves Equation 14. ■

A.2 Token decomposition of FL and SLD

Write $\ell(\theta, y_j \mid x, y_{<j}) := -\log p_\theta(y_j \mid x, y_{<j})$ for token loss. For every contiguous target span $y = (y_1, \ldots, y_m)$,

$$FL(y \mid x) = \sum_{j=1}^{m} \ell(\theta_T, y_j \mid x, y_{<j}), \tag{30}$$

$$\mathrm{SLD}_{t_r}(y \mid x) = \sum_{j=1}^{m} \left[\ell(\theta_{t_r}, y_j \mid x, y_{<j}) - \ell(\theta_T, y_j \mid x, y_{<j})\right]. \tag{31}$$

Interpretation. The first identity says that total span FL retains the full tokenwise loss mass; equivalently, it is m times mean span FL. Mean FL must therefore be multiplied by the span length before using the unbarred probability identity in Appendix A.1. The second identity says that tokenwise checkpoint changes add with their signs: improvements increase span SLD, while regressions reduce it. Thus a contiguous run can be aggregated without approximation. Additivity alone does not select a threshold, minimum run length, or boundary estimator, so the TSL design choices and localization accuracy remain empirical questions.

Proof. Equation 30 is the definition of total autoregressive span loss. For SLD, subtract the final total loss from the reference total loss and distribute subtraction across the finite sum:

$$\begin{aligned}\mathrm{SLD}_{t_r}(y \mid x) &= L_{t_r}(y \mid x) - L_T(y \mid x) \\ &= \sum_{j=1}^{m} \ell(\theta_{t_r}, y_j \mid x, y_{<j}) - \sum_{j=1}^{m} \ell(\theta_T, y_j \mid x, y_{<j}) \\ &= \sum_{j=1}^{m} \left[\ell(\theta_{t_r}, y_j \mid x, y_{<j}) - \ell(\theta_T, y_j \mid x, y_{<j})\right].\end{aligned} \tag{32}$$

This proves both displayed identities. ■

A.3 TSL AS A GREEDY TOKEN-THRESHOLD IMPLEMENTATION

We now formalize the connection between the span identities and Algorithm 1. For a generated sequence, write

$$f_t := -\log p_{\theta_T}(X_t \mid X_{<t}), \tag{33}$$

$$g_t := -\log p_{\theta_{t_r}}(X_t \mid X_{<t}) + \log p_{\theta_T}(X_t \mid X_{<t}). \tag{34}$$

For an FL threshold ζ, call an interval $R = [a, b]$ pointwise feasible if $f_t \le \zeta$ for every $t \in R$. For an SLD threshold ζ, call it pointwise feasible if $g_t \ge \zeta$ for every $t \in R$. Let $|R| = b - a + 1$, and retain only intervals with $|R| \ge L_{\min}$, where $L_{\min} \ge 1$ is an integer. For FL assume $\zeta > 0$, so the reproduction level $e^{-m\zeta}$ lies in $(0, 1)$.

Proposition A.1 (Greedy maximal-run characterization)**.** *For either score direction, Algorithm 1 enumerates all maximal pointwise-feasible intervals in one left-to-right scan and returns a solution of*

$$\widehat{R} \in \arg\max_R |R| \quad \textit{subject to } R \textit{ being pointwise feasible and } |R| \ge L_{\min}. \tag{35}$$

If no feasible interval exists, the algorithm returns $\varnothing$. Otherwise write $\widehat{R} = [\widehat{s}, \widehat{e}]$. If the returned interval has length m and continuation $y = X_{\widehat{R}}$, then FL thresholding guarantees

$$FL(y \mid X_{<\widehat{s}}) \le m\zeta, \qquad p_{\theta_T}(y \mid X_{<\widehat{s}}) \ge e^{-m\zeta}, \tag{36}$$

whereas SLD thresholding guarantees

$$\mathrm{SLD}_{t_r}(y \mid X_{<\widehat{s}}) \ge m\zeta, \qquad \frac{p_{\theta_T}(y \mid X_{<\widehat{s}})}{p_{\theta_{t_r}}(y \mid X_{<\widehat{s}})} \ge e^{m\zeta}. \tag{37}$$

If $y \in \mathcal{C}_m(S)$ and $m \ge \gamma$, the FL result is Proposition 4.2 with the length-dependent level $\tau_m = e^{-m\zeta}$. The SLD result certifies probability amplification; it also yields the proposition's probability certificate at every $\tau \le p_{\theta_{t_r}}(y \mid X_{<\widehat{s}})e^{m\zeta}$.

Proof. Fix either score direction and let $A \subseteq \{1, \dots, N\}$ be the flagged positions. Thus $A = \{t : f_t \le \zeta\}$ for FL and $A = \{t : g_t \ge \zeta\}$ for SLD. A maximal contiguous run of A is an interval $[a, b]$ such that every $t \in [a, b]$ belongs to A, while $a - 1 \notin A$ when $a > 1$ and $b + 1 \notin A$ when $b < N$.

During the left-to-right scan, the algorithm opens a run exactly when it encounters a flagged position whose predecessor is unflagged or absent. It keeps that run open while positions remain flagged and closes it exactly at the first subsequent unflagged position, or after position N. Hence each recorded interval is a maximal run. Conversely, the scan must encounter the left endpoint of every maximal run, so no maximal run is omitted. Discarding runs shorter than $L_{\min}$ leaves precisely the eligible maximal runs.

Every pointwise-feasible interval R is contained in a unique maximal run $R^{\max}$. Consequently $|R| \le |R^{\max}|$. At least one longest feasible interval is therefore a maximal run, and selecting the longest surviving maximal run solves Equation 35. Ties are resolved by the smallest start index, as in Algorithm 1, so the returned interval is unique.

Now suppose FL is used and $\widehat{R} = [\widehat{s}, \widehat{e}]$ has length m. Every $t \in \widehat{R}$ satisfies $f_t \le \zeta$, so finite-sum monotonicity and Equation 30 give

$$FL(y \mid X_{<\widehat{s}}) = \sum_{t=\widehat{s}}^{\widehat{e}} f_t \le \sum_{t=\widehat{s}}^{\widehat{e}} \zeta = m\zeta. \tag{38}$$

Exponentiating its negative and using Equation 15 yields $p_{\theta_T}(y \mid X_{<\widehat{s}}) \ge e^{-m\zeta}$. This proves Equation 36 and its main-text form, Equation 11.

For SLD, every $t \in \widehat{R}$ satisfies $g_t \ge \zeta$. Therefore Equation 31 gives

$$\mathrm{SLD}_{t_r}(y \mid X_{<\widehat{s}}) = \sum_{t=\widehat{s}}^{\widehat{e}} g_t \ge \sum_{t=\widehat{s}}^{\widehat{e}} \zeta = m\zeta. \tag{39}$$

Exponentiating and applying Equation 16 proves Equation 37 and Equation 12.

Finally, if y is indexed and $m \ge \gamma$, set $\tau_m = e^{-m\zeta}$. The FL inequality is exactly $FL(y \mid X_{<\widehat{s}}) \le -\log \tau_m$, so Proposition 4.2 applies. The analogous SLD statement follows by multiplying its likelihood-ratio bound by the reference probability. This completes the proof. ■

Scope of the connection. The algorithm is greedy only for the pointwise-threshold longest-run problem; it is not an optimizer of total FL over all possible spans. In particular, $e^{-m\zeta}$ decreases with m for $\zeta > 0$, so choosing the longest run does not claim that it has the greatest whole-span probability. Length is used as a localization prior aligned with the minimum verbatim-match length and is validated empirically by IoU and AUC. Conversely, a span can satisfy the total FL or SLD condition in Proposition 4.2 even when one token fails the pointwise threshold; the TSL rule is sufficient but not exhaustive. To impose one fixed reproduction level τ across different lengths, one must directly test total FL against $-\log\tau$ (equivalently use the length-dependent mean threshold $-\log\tau/m$). Finally, neither score condition proves $y \in \mathcal{C}_m(S)$: indexed membership is established by the corpus-matching stage of the experiments.

A.4 Counterfactual memorization and the exact SLD decomposition

Full definition. Following Feldman (2020), and its language-model adaptation in Zhang et al. (2023), let $\mathcal{D}$ be a training corpus containing the indexed record z_i. Let the possibly randomized training procedure produce the final included model θ_T^{in} from $\mathcal{D}$ and the final omitted model θ_T^{out} from $\mathcal{D} \setminus \{z_i\}$. The two training worlds use the same architecture, optimization procedure, data construction, and evaluation loss; they differ only in whether z_i is included. Expectations below are over all training randomness. For a fixed target loss $\ell(\theta, z_i)$, define

$$I_i := \mathbb{E}[\ell(\theta_T^{\text{in}}, z_i)], \qquad \text{included final mean loss,} \tag{40}$$

$$O_i := \mathbb{E}[\ell(\theta_T^{\text{out}}, z_i)], \qquad \text{omitted final mean loss.} \tag{41}$$

For a token sequence, we take ℓ to be mean per-token negative log-likelihood, matching the empirical proxy. The loss-based counterfactual memorization of z_i is

$$\text{mem}_{\text{loss}}(z_i) := O_i - I_i. \tag{42}$$

Thus $\text{mem}_{\text{loss}}(z_i) > 0$ means that including z_i lowers its expected final mean loss; larger values indicate a larger record-specific training effect under this loss convention.

Now choose a reference checkpoint $t_r < T$ from the included training run and define

$$R_i := \mathbb{E}[\ell(\theta_{t_r}^{\text{in}}, z_i)]. \tag{43}$$

By the definition of mean SLD and linearity of expectation,

$$\mathbb{E}[\overline{\text{SLD}}_{t_r}(z_i)] = R_i - I_i. \tag{44}$$

Therefore,

$$\mathbb{E}[\overline{\text{SLD}}_{t_r}(z_i)] + (O_i - R_i) = (R_i - I_i) + (O_i - R_i) \tag{45}$$

$$= O_i - I_i \tag{46}$$

$$= \text{mem}_{\text{loss}}(z_i). \tag{47}$$

Writing $c_i^{\text{ref}} := O_i - R_i$ yields Equation 6. The correction compares the unobserved omitted final model with the observed included reference checkpoint. Equivalently, $\mu_i^{\text{ref}} := R_i - O_i = -c_i^{\text{ref}}$ is the learning-compatible-with-omission term, and $\text{mem}_{\text{loss}}(z_i) = \mathbb{E}[\overline{\text{SLD}}_{t_r}(z_i)] - \mu_i^{\text{ref}}$. No smoothness, stability, bounded-loss, or optimizer assumption is used.

This identity is stated for the loss-based counterfactual score. The accuracy-gap score used as the primary empirical target in Table 8 admits no analogous exact decomposition, since it is not an expectation of the same evaluation loss; the correlations reported against it are empirical.

A.5 Supporting causal bounds and proof of Theorem 4.4

Formal assumptions. The theorem uses the following conditions. (A1) The target record z_i, prefix x, continuation y, reference checkpoint t_r, and decoder distribution are fixed before calibration, and the decoder assigns positive probability to y in both training worlds. (A2) The included and omitted potential training worlds use the same architecture, optimizer, data construction, evaluation loss, and decoding coupling; they differ only in whether z_i is included. (A3) The descriptor, matching rule, checkpoint-selection rule, and K excluded controls are pre-specified without inspecting the target

or control changes. (A4) Conditional on the matching information, either $\bar{U}_1, \ldots, \bar{U}_K, \bar{\mu}^{\mathrm{ref}}_{i,\mathrm{pair}}$ are exchangeable, or there exists an exchangeable auxiliary control $\bar{U}_{K+1}$ satisfying $\bar{\mu}^{\mathrm{ref}}_{i,\mathrm{pair}} \le \bar{U}_{K+1}$ almost surely. The latter is the conservative matched-control condition used in the main text. These conditions concern calibration validity; the algebraic FL, SLD, PN, and PNS bounds below do not depend on the optimizer. In (A4), the auxiliary control must be jointly exchangeable with all K observed controls. Matching descriptors alone does not imply (A4), since the observed controls use the included model whereas the target change uses an omitted model. We write $\Pr_{\mathrm{cal}}$ for the joint law of this target/control array conditional on the specified matching information. This is marginal coverage over that law, which can include training randomness; it is not coverage conditional on the realized model pair, target change, or observed calibration values unless (A4) also holds under that conditioning.

For any span v, write $\bar{L}_t(v \mid u) := L_t(v \mid u)/|v|$. For a matched excluded span $(\widetilde{x}_j, \widetilde{y}_j)$, define its observed mean checkpoint change by

$$\bar{U}_j := \bar{L}_{t_r}(\widetilde{y}_j \mid \widetilde{x}_j) - \bar{L}_T(\widetilde{y}_j \mid \widetilde{x}_j), \qquad j = 1, \ldots, K. \tag{48}$$

For $K \ge 1$ and $1/(K+1) \le \delta < 1$, sort these changes and set

$$k_\delta := \lceil (K+1)(1-\delta) \rceil, \qquad \bar{b}_\delta := \bar{U}_{(k_\delta)}, \tag{49}$$

where $\bar{U}_{(k)}$ is the kth smallest value. Thus $\bar{b}_\delta$ is a finite-sample one-sided upper calibration bound. In particular, $K \ge 19$ is needed to use $\delta = 0.05$. If $0 < \delta < 1/(K+1)$, define $\bar{b}_\delta = +\infty$, giving only the trivial zero lower bounds. Here "mean" refers to per-token normalization: $\bar{b}_\delta$ bounds one target's omitted-world change, not the population mean change.

Potential-outcome and calibration details. Fix a record z_i, prefix x, and exact target continuation $y = (y_1, \ldots, y_m)$. Intervention $a = 1$ includes z_i in training and intervention $a = 0$ omits it, with all other aspects of training matched. Write $L^a_t(y \mid x)$ for total loss and $\bar{L}^a_t(y \mid x) := L^a_t(y \mid x)/m$ for mean per-token loss at checkpoint t under intervention a. For the paired-model identities, condition on the realized included/omitted training pair (including its reference checkpoint). Then the losses are fixed, and p_a denotes the probability over decoding randomness alone. This conditioning is suppressed in the notation. If training randomness were instead averaged into p_a, the correct expression would be $p_a = \mathbb{E}[e^{-L^a_T(y|x)}]$, not $e^{-\mathbb{E}[L^a_T(y|x)]}$. The expected-loss identity in Appendix A.4 therefore does not itself imply these paired probability identities. For a paired decoding unit ω containing the shared decoding randomness, let $X^a_{1:m}(\omega)$ be the first m tokens generated after x and define

$$H_i(a; \omega) := \mathbb{I}\{X^a_{1:m}(\omega) = y\}. \tag{50}$$

Write

$$p_a := \Pr\{H_i(a; \omega) = 1\}, \qquad a \in \{0, 1\}, \tag{51}$$

for the hit probability under intervention a, taken over decoding randomness with the training pair held fixed. Thus the estimand concerns one specified continuation after one specified prefix, rather than an unspecified match in a free-running generation. Monotonicity means $H_i(1) \ge H_i(0)$ almost surely: inclusion never destroys a hit that would have occurred under omission. The lower bounds below do not require this assumption; monotonicity only makes them exact.

Before observing the target mean SLD or calibration changes, fix a descriptor ϕ and matching rule. A matched excluded span $(\widetilde{x}_j, \widetilde{y}_j)$ has a designated source record that was not used in training and pre-specified properties, for example span length, prefix length, mean reference loss, token frequency, duplication statistics, and source domain, matched to the target. Target and controls use the same loss normalization, prefix convention, decoder distribution, checkpoint pair, and checkpoint-selection rule. Their mean changes $\bar{U}_j$ are defined in Equation 48. The target's unobserved omitted-world improvement is

$$\bar{\mu}^{\mathrm{ref}}_{i,\mathrm{pair}} := \bar{L}^1_{t_r}(y \mid x) - \bar{L}^0_T(y \mid x). \tag{52}$$

Conditional on the matching information, valid calibration requires either $\bar{U}_1, \ldots, \bar{U}_K, \bar{\mu}^{\mathrm{ref}}_{i,\mathrm{pair}}$ to be exchangeable, or an auxiliary $\bar{U}_{K+1}$ such that $\bar{U}_1, \ldots, \bar{U}_K, \bar{U}_{K+1}$ are exchangeable and $\bar{\mu}^{\mathrm{ref}}_{i,\mathrm{pair}} \le \bar{U}_{K+1}$ almost surely. The second condition formalizes conservative calibration: the controls upper-represent how much the target could have improved without its indexed record.

Finally, define the paired proxy quantities

$$\bar{F}_i := \bar{L}_T^1(y \mid x), \tag{53}$$

$$\bar{S}_i := \bar{L}_{t_r}^1(y \mid x) - \bar{L}_T^1(y \mid x), \tag{54}$$

and, for $p_0 > 0$, the paired log-probability effect

$$\Delta_i(y \mid x) := \log \frac{p_1}{p_0}. \tag{55}$$

A.5.1 Sharp paired-model bounds

Proposition A.2 (Sharp paired-model bounds)**.** *For any joint coupling of the potential outcomes in Definition 4.3 with $p_1 > 0$, the marginal hit probabilities give the sharp bounds*

$$\left[1 - \frac{p_0}{p_1}\right]_+ \le \mathrm{PN}_i \le \min\left\{1, \frac{1-p_0}{p_1}\right\}, \tag{56}$$

$$[p_1 - p_0]_+ \le \mathrm{PNS}_i \le \min\{p_1, 1-p_0\}, \tag{57}$$

where $[u]_+ := \max\{u, 0\}$. Moreover

$$p_1 = e^{-m\bar{F}_i}, \qquad \Delta_i(y \mid x) = m(\bar{S}_i - \bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}}), \tag{58}$$

and consequently

$$\mathrm{PN}_i \ge \left[1 - \exp\left\{-m(\bar{S}_i - \bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}})\right\}\right]_+, \tag{59}$$

$$\mathrm{PNS}_i \ge e^{-m\bar{F}_i}\left[1 - \exp\left\{-m(\bar{S}_i - \bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}})\right\}\right]_+, \tag{60}$$

$$\mathrm{PN}_i \ge [1 - e^{-\Delta_i(y|x)}]_+, \tag{61}$$

$$\mathrm{PNS}_i \ge p_1[1 - e^{-\Delta_i(y|x)}]_+. \tag{62}$$

Under monotonicity these lower bounds are equalities; without monotonicity they remain valid but need not be exact.

Proof. For $a, b \in \{0, 1\}$, set

$$\pi_{ab} := \Pr\{H_i(1) = a, H_i(0) = b\}. \tag{63}$$

The four cell probabilities are nonnegative and sum to one. Their marginals satisfy

$$p_1 = \pi_{10} + \pi_{11}, \tag{64}$$

$$p_0 = \pi_{01} + \pi_{11}. \tag{65}$$

By definition, $\mathrm{PNS}_i = \pi_{10}$. Subtracting the marginal identities gives

$$p_1 - p_0 = \pi_{10} - \pi_{01}. \tag{66}$$

Since $\pi_{01}, \pi_{10} \ge 0$,

$$\pi_{10} \ge [p_1 - p_0]_+. \tag{67}$$

Also, $\pi_{10} \le p_1$ by Equation 64, and $\pi_{10} \le 1 - p_0$ because $\{H_i(1) = 1, H_i(0) = 0\} \subseteq \{H_i(0) = 0\}$. Hence

$$\pi_{10} \le \min\{p_1, 1 - p_0\}, \tag{68}$$

which proves Equation 57. These bounds are sharp: for any π_{10} in this interval, set $\pi_{11} = p_1 - \pi_{10}$, $\pi_{01} = p_0 - \pi_{11}$, and let π_{00} be the remaining mass. The interval endpoints ensure that all four cells are nonnegative and have the required marginals.

Since $p_1 > 0$, conditional probability gives

$$\mathrm{PN}_i = \frac{\Pr\{H_i(1) = 1, H_i(0) = 0\}}{\Pr\{H_i(1) = 1\}} = \frac{\pi_{10}}{p_1}. \tag{69}$$

Dividing the sharp PNS interval by p_1 proves Equation 56 and its sharpness.

If $p_0 > 0$, Equation 55 gives

$$\frac{p_0}{p_1} = e^{-\Delta_i(y|x)}. \tag{70}$$

Substitution into the lower bounds proves Equations 61 and 62.

For the fixed exact-continuation event, Equation 15 in Appendix A.1 gives

$$p_1 = e^{-L_T^1(y|x)} = e^{-m\bar{F}_i}. \tag{71}$$

Using Equation 54 and Equation 52,

$$\begin{aligned} m(\bar{S}_i - \bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}}) &= m\big(\bar{L}_{t_r}^1 - \bar{L}_T^1\big) - m\big(\bar{L}_{t_r}^1 - \bar{L}_T^0\big) \\ &= L_T^0(y \mid x) - L_T^1(y \mid x) \\ &= \log \frac{p_1}{p_0} = \Delta_i(y \mid x). \end{aligned} \tag{72}$$

Together with Equation 71, this proves Equation 58 and hence Equations 59 and 60.

Finally, monotonicity $H_i(1) \geq H_i(0)$ rules out π_{01}. Hence Equation 66 gives $\pi_{10} = p_1 - p_0$, making the PNS lower bound exact; Equation 69 makes the PN lower bound exact as well. ■

A.5.2 Rank-calibrated single-run certificate

The target's omitted-world change

$$\bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}} = \bar{L}_{t_r}^1(y \mid x) - \bar{L}_T^0(y \mid x) \tag{73}$$

is unavailable from the included run. Define the calibrated margin

$$\bar{d}_{i,\delta} := \big[\bar{S}_i - \bar{b}_\delta\big]_+ . \tag{74}$$

Under exact conditional exchangeability, the randomized rank of $\bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}}$ among $\bar{U}_1, \ldots, \bar{U}_K, \bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}}$ is uniform. Under the conservative alternative, the same statement holds for the auxiliary $\bar{U}_{K+1}$ and $\bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}} \leq \bar{U}_{K+1}$ almost surely. If the reference checkpoint or target span is selected adaptively, the selection procedure must preserve (A4), for example through an independent selection split and valid conditional matching. Applying the same rule only to the observed controls does not establish exchangeability with the unobserved target change. Merely saving more checkpoints does not increase confidence.

Proposition A.3 (Rank-calibrated single-run certificate)**.** *Under (A1)–(A4) with $K \geq 1$ and $1/(K+1) \leq \delta < 1$,*

$$\Pr_{\mathrm{cal}}\big\{\bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}} \leq \bar{b}_\delta\big\} \geq 1 - \delta, \tag{75}$$

and on the same event the reported lower bounds hold simultaneously:

$$\mathrm{PN}_i \geq 1 - e^{-m\bar{d}_{i,\delta}}, \tag{76}$$

$$\mathrm{PNS}_i \geq e^{-m\bar{F}_i}\left(1 - e^{-m\bar{d}_{i,\delta}}\right). \tag{77}$$

Proof. Under exact exchangeability, the rank of $\bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}}$ among the $K+1$ values is uniform on $\{1, \ldots, K+1\}$ when ties are broken by independent randomization. If that rank is at most k_δ, the target value cannot exceed the k_δth smallest of the K controls: otherwise at least k_δ controls would be strictly smaller and its rank would be larger. Thus the rank event is contained in the coverage event. Ties can only make the non-strict coverage event larger. Therefore

$$\begin{aligned} \Pr_{\mathrm{cal}}\big\{\bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}} \leq \bar{U}_{(k_\delta)}\big\} &\geq \frac{k_\delta}{K+1} \\ &\geq 1 - \delta, \end{aligned} \tag{78}$$

where the last inequality follows from $k_\delta = \lceil (K+1)(1-\delta) \rceil$. Under the conservative alternative, exchangeability gives the same bound with $\bar{U}_{K+1}$ in place of $\bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}}$; because $\bar{\mu}_{i,\mathrm{pair}}^{\mathrm{ref}} \leq \bar{U}_{K+1}$ almost

surely, the event $\{\bar{U}_{K+1} \le \bar{U}_{(k_\delta)}\}$ implies $\{\bar{\mu}^{\mathrm{ref}}_{i,\mathrm{pair}} \le \bar{U}_{(k_\delta)}\}$. Hence the same coverage bound holds. Since $\bar{b}_\delta = \bar{U}_{(k_\delta)}$, define

$$\mathcal{E}_\delta := \{\bar{\mu}^{\mathrm{ref}}_{i,\mathrm{pair}} \le \bar{b}_\delta\}. \tag{79}$$

Equation 78 gives $\Pr_{\mathrm{cal}}(\mathcal{E}_\delta) \ge 1 - \delta$.

On $\mathcal{E}_\delta$, Equation 58 gives

$$\begin{aligned}\Delta_i(y \mid x) &= m(\bar{S}_i - \bar{\mu}^{\mathrm{ref}}_{i,\mathrm{pair}}) \\ &\ge m(\bar{S}_i - \bar{b}_\delta).\end{aligned} \tag{80}$$

If $\bar{S}_i \le \bar{b}_\delta$, then $\bar{d}_{i,\delta} = 0$ by Equation 74 and both reported lower bounds are zero, so they are valid trivially. If $\bar{S}_i > \bar{b}_\delta$, then $\bar{d}_{i,\delta} = \bar{S}_i - \bar{b}_\delta > 0$ and Equation 80 implies $\Delta_i(y \mid x) \ge m\bar{d}_{i,\delta}$. Because $u \mapsto 1 - e^{-u}$ is increasing, Proposition A.2 yields

$$\mathrm{PN}_i \ge 1 - e^{-m\bar{d}_{i,\delta}}, \tag{81}$$

$$\mathrm{PNS}_i \ge e^{-m\bar{F}_i}(1 - e^{-m\bar{d}_{i,\delta}}). \tag{82}$$

Both inequalities hold on the same event $\mathcal{E}_\delta$, whose $\Pr_{\mathrm{cal}}$-probability is at least $1 - \delta$. No extra factor of two is needed for PN and PNS because the same calibration event implies both bounds for this target. This proves Equation 8. ■

Worked LLM calculation. Consider one fixed $m = 50$ token span under raw ancestral decoding. Suppose its reference and final mean losses are $\bar{L}^1_{t_r}(y \mid x) = 0.09$ and $\bar{F}_i = \bar{L}^1_T(y \mid x) = 0.02$ nats per token, so $\bar{S}_i = 0.07$. With $K = 19$ excluded controls and $\delta = 0.05$, $k_\delta = \lceil 20(0.95) \rceil = 19$ and $\bar{b}_\delta$ is the largest calibration drift. If $\bar{b}_\delta = 0.02$, then $\bar{d}_{i,\delta} = 0.05$ and

$$\mathrm{PN}_i \ge 1 - e^{-50(0.05)} \approx 0.918, \tag{83}$$

$$\mathrm{PNS}_i \ge e^{-50(0.02)}(1 - e^{-50(0.05)}) \approx 0.338. \tag{84}$$

Thus the procedure certifies with at least 95% coverage that at least 91.8% of included-world hits require the record and that inclusion creates at least 33.8 percentage points of exact-span hit probability. The smaller PNS bound reflects the final reproduction probability $e^{-50(0.02)} = e^{-1} \approx 0.368$.

The confidence statement has the repeated-calibration interpretation

$$\Pr_{\mathrm{cal}}\{\text{reported lower bounds are valid}\} \ge 1 - \delta. \tag{85}$$

It does not assign probability δ to the calibration assumption being wrong. If one pre-specified calibration set is reused for every target, its $\bar{b}_\delta$ is fixed after calibration. Conservativeness for each target gives per-target coverage, not simultaneous coverage. For J pre-specified targets with respective error levels δ_i, a union bound gives

$$\Pr_{\mathrm{cal}}\left\{\bigcap_{i=1}^{J} \mathcal{E}_{i,\delta_i}\right\} \ge 1 - \sum_{i=1}^{J} \delta_i. \tag{86}$$

Thus $\delta_i = \delta/J$ suffices for joint coverage $1 - \delta$; independence between targets is not required, even when controls are reused. A finite rank bound at this level requires $1/(K+1) \le \delta/J$. Alternatively, a shared threshold must be calibrated to dominate the maximum target change jointly. A threshold tuned for localization AUC is not such a bound.

Remark A.4 (Decoder and event scope). The probability identities are exact for raw ancestral sampling from p_θ: each next token is drawn from the model's full softmax distribution. For temperature, top-k, or top-p sampling, the identities hold after replacing p_θ by the decoder's modified distribution and recomputing decoder-adjusted FL and SLD, provided the target span has positive probability under each distribution being compared. Our experiments use raw-model token FL/SLD for localization and their mean-span aggregates for AUC while generating with a modified decoder; effectiveness under that decoder is therefore empirical rather than an exact decoder-probability identity. Greedy decoding is deterministic and has no nontrivial sampling-probability interpretation.

B Qualitative Examples and Algorithmic Details

B.1 Qualitative Examples

We present additional qualitative examples of memorized content produced by the model. Each example corresponds to a generated sequence of at most 512 tokens. For each sequence, we highlight the contiguous text span that is reproduced verbatim from the training data.

Example A: The Federal Aviation Administration is warning airlines not to place too many seats at any one time on a domestic flight or take off on a long-haul flight from Washington. The Transportation Security Administration on Tuesday said there are more than 4,800 flights scheduled from and to the District of Columbia on American Airlines, Delta Air Lines, United Airlines, and Continental Airlines between 6 a.m. and 10 p.m. every day this summer. “TSA continues to track this risk closely and advises airlines to take necessary precautions and to make the most of available travel options,” the agency said in a statement. “Travelers should use common sense when selecting travel options and avoid placing more than one seat in a row on flights from and to the nation’s capital.” On a domestic flight, passengers who board first on a flight from a hub airport to an urban area will get a free checked bag as well as food or drinks. If passengers get a seat in the first group, the bag will be checked. But if there is a second group that is assigned seats that don’t require a bag, passengers on the first group will be given a free checked bag. “No food or drink may be given out until the flight is over,” the TSA said. TSA notes that passengers who try to travel alone on domestic flights from Washington or any major U.S. city are advised to contact their airline and arrange a solo flight. TSA advises those who aren’t traveling alone to check with their airline for the best seats, since they often have less leg room than people who share a seat with others. TSA also recommends travelers try to fly on an airline that has both a “D” rating and the American equivalent of a “B” rating, for which you don’t need to be traveling within an hour of your destination, or any flight within eight hours. You May Also Be Interested In These Stories(TM and © Copyright 2014 CBS Radio Inc. and its relevant subsidiaries. CBS RADIO and EYE Logo TM and Copyright 2014 CBS Broadcasting Inc. Used under license. All Rights Reserved. This material may not be published, broadcast, rewritten, or redistributed. The Associated Press contributed to this report.)",

Example B:
The U.S.S. Mariner is in no way affiliated with, condoned or given any notice by the Seattle Mariners baseball team, who have their own website . Similarly, we have no association with the ownership group or any businesses related to the Mariners. All article text is written by the authors, all pictures are taken by the authors, who retain copyright to their works. No copying or reproduction of any content here, photographic or otherwise, is authorized. Please email us if you wish to reproduce our work. Matt Adams is the second-base coach in the Texas Rangers’ Triple-A affiliate, where in 2011 he played for the O-Hos and the Rangers’ Single-A Fort Wayne squad. That team finished fourth in the Midwest League in batting. Matt played a role in helping the Triple-A club to its first win of the season on Wednesday, beating Fort Wayne, 13-10.After the game, Matt and I sat down with Matt Adams on the phone about his time in Fort Wayne, and his relationship with the Fort Wayne baseball club.GQ: How did you get into baseball?Adams: I grew up watching baseball—I was a big baseball fan. I grew up in Houston, Texas, so I always had a soft spot for baseball. I was the middle of three sons and my eldest son was in college, so I grew up watching baseball. My dad and my mom both played, so that was a big part of the family’s life. Baseball was the sport I grew up watching, so I knew it was cool. I got into football from there, so football and baseball were pretty natural for me. Then college ball came into play and I was hooked. I really enjoyed college, because I played every other weekend and it was really nice and baseball, but when I decided to start playing baseball, I did not know anything about it. When I started, I was really just playing for fun, and I didn’t even know much about it. I had just played with my buddies, so baseball was like a big part of my life. The sport took off after college and it really just took off after that.GQ: How did you feel playing in Double-A and Triple-A?Adams: I always felt good as far as hitting and pitching. You always felt confident

in your ability to go out there and pitch every time. You never felt like you had to do the same thing",

Example C: The House GOP is expected to pass its version of the American Health Care Act next week, setting up a vote on that version next week by the Senate — a delay that President Trump has previously predicted he would like to see happen. The House is expected to vote on the American Health Care Act on Thursday, with the Senate expected to take up the bill next Tuesday. ADVERTISEMENT The measure was first unveiled in March, with its first full page of text released by Rep. Tom MacArthur (R-N.J.) on March 23. MacArthur predicted at the time that the House health bill would be d̈ead on arrivalïn the Senate. He noted that the Senate's version of the legislation did not include measures to address the opioid epidemic. "In fact, the Senate's plan includes no efforts to address prescription painkiller abuse, is $300 billion lower than our proposal, and could cost more than double. There are no Medicare changes. "There are not even any provisions to help us pay for it," MacArthur said in a conference call with reporters on Wednesday. Senate Majority Leader Mitch McConnell Addison (Mitch) Mitchell McConnellHouse to push back at Trump on border Democrats block abortion bill in Senate Overnight Energy: Climate protesters storm McConnell's office | Center-right group says Green New Deal could cost trillion | Dire warnings from new climate studies MORE (R-Ky.) has said he is confident the Senate will take up the House measure and pass it as is, without tweaks. But Democrats and centrists have said they are concerned the House bill, which would roll back major portions of ObamaCare, would result in more people losing coverage and less affordable healthcare. The Senate Finance Committee will hold a hearing on the House legislation next week. The panel voted on Monday to send the measure, which aims to repeal ObamaCare, to the full Senate for a vote.",

B.2 Algorithms

We summarize our token span localization procedure in Algorithm 2. Given a generated token sequence and a per-token memorization score (e.g., SLD or FL), the algorithm identifies contiguous token spans whose scores cross a fixed threshold (ζ). Token spans shorter than a minimum length are discarded, and among the remaining candidates, the longest span is selected as the predicted memorized span. This procedure is used consistently across all experiments and qualitative analyses in the paper.

Algorithm 3 summarizes the AUC evaluation procedure used for TSL. Here $y^{(j)} = 1$ denotes an indexed verbatim hit, not training membership. The span-crossing rule uses $\leq \zeta$ for FL and $\geq \zeta$ for SLD. For FL, lower sequence scores indicate hits and S_{default} must exceed all retained-span scores; for SLD the directions reverse. The span threshold ζ and minimum length $L_{\min}$ are selected on a validation set and then frozen; the decision threshold of the second stage is not a tuned hyperparameter but is swept implicitly as part of the ROC computation. Both positive and negative examples are required for ROC AUC to be defined. With evidence-oriented scores $Q_j = S^{(j)}$ for the higher direction and $Q_j = -S^{(j)}$ for the lower direction, the same AUC is

$$\text{AUC} = \frac{1}{M_+ M_-} \sum_{j:y^{(j)}=1} \sum_{k:y^{(k)}=0} \left(\mathbb{I}[Q_j > Q_k] + \tfrac{1}{2}\mathbb{I}[Q_j = Q_k] \right), \tag{87}$$

where M_+ and M_- count positives and negatives. This also specifies the treatment of ties, including multiple no-span examples. Reducing each example to a single scalar keeps TSL directly comparable to prior membership inference methods, which produce one score per example.

C Experimental Setup and Compute

C.1 Generation and Evaluation Protocol

Table 4 is the authoritative record of how many sequences were generated for each model and how many of them contain a verbatim hit. All other tables in the paper draw their sample counts from it.

Algorithm 2 Token Span Localization (TSL)

Input: token sequence $z_{1:N}$; per-token scores $\mathrm{score}(z_i)$ (either SLD or FL); threshold ζ; minimum span length $L_{\min}$
Output: inclusive predicted span $[\text{start}_{\text{pred}}, \text{end}_{\text{pred}}]$ or $\varnothing$
Definition: $\mathrm{isMemorized}(z_i; \zeta)$ indicates whether z_i crosses the threshold. For SLD, this corresponds to $\mathrm{score}(z_i) \geq \zeta$, while for FL it corresponds to $\mathrm{score}(z_i) \leq \zeta$. This is a score flag, not a corpus-membership certificate.
Initialize empty list of spans $\mathcal{S} \leftarrow \emptyset$
$inSpan \leftarrow$ **false**
for $i = 1$ **to** N **do**
  if $\mathrm{isMemorized}(z_i; \zeta)$ **then**
    if not $inSpan$ **then**
      $start \leftarrow i$
      $inSpan \leftarrow$ **true**
    end if
  else
    if $inSpan$ **then**
      $end \leftarrow i - 1$
      if $end - start + 1 \geq L_{\min}$ **then**
        Add span $(start, end)$ to $\mathcal{S}$
      end if
      $inSpan \leftarrow$ **false**
    end if
  end if
end for
if $inSpan$ **then**
  $end \leftarrow N$
  if $end - start + 1 \geq L_{\min}$ **then**
    Add span $(start, end)$ to $\mathcal{S}$
  end if
end if
if $\mathcal{S} = \emptyset$ **then**
  return $\varnothing$
else
  Select longest span $(s^*, e^*) \in \arg\max_{(s,e)\in\mathcal{S}}(e - s + 1)$, breaking ties by the smallest start index
  return (s^*, e^*)
end if

Algorithm 3 Evaluating TSL-based verbatim-hit detection (fixed ζ, $L_{\min}$; AUC over decision thresholds)

Input: examples $\{(z^{(j)}_{1:N_j}, y^{(j)})\}_{j=1}^{M}$, $y^{(j)} \in \{0,1\}$; per-token scores $\text{score}_i^{(j)}$ (SLD or FL); span threshold ζ (chosen on validation); minimum span length $L_{\min}$ (chosen on validation); span direction $dir_{\text{span}} \in \{\texttt{below}, \texttt{above}\}$; sequence direction $dir_{\text{seq}} \in \{\texttt{higher}, \texttt{lower}\}$; finite no-span score S_{default} ranked below every retained-span score in evidence for a hit
Output: ROC AUC
// Step 1: convert token-level scores to one sequence-level score per example
for $j = 1$ **to** M **do**
    Find all maximal contiguous spans $\mathcal{S}^{(j)}$ such that $\text{Cross}(\text{score}_i^{(j)}, \zeta, dir_{\text{span}})$ holds for all i in the span
    Use inclusive endpoints; remove any $(s, e) \in \mathcal{S}^{(j)}$ with $(e - s + 1) < L_{\min}$
    if $\mathcal{S}^{(j)} = \emptyset$ **then**
        $S^{(j)} \leftarrow S_{\text{default}}$ {no detected span}
    else
        $(s^*, e^*) \leftarrow$ longest span in $\mathcal{S}^{(j)}$, breaking ties by the smallest start index
        $S^{(j)} \leftarrow \dfrac{1}{e^* - s^* + 1} \sum_{i=s^*}^{e^*} \text{score}_i^{(j)}$ {aggregate span to a scalar}
    end if
end for
// Step 2: compute ROC by sweeping a decision threshold over $\{S^{(j)}\}$
Define decision-threshold set $\mathcal{T} \leftarrow \{S^{(j)}\}_{j=1}^{M}$ {e.g., sorted unique values}
for each $\xi \in \mathcal{T}$ **do**
    if $dir_{\text{seq}} = \texttt{higher}$ **then**
        $\hat{y}^{(j)}(\xi) \leftarrow \mathbb{I}[S^{(j)} \geq \xi]$ for all j
    else
        $\hat{y}^{(j)}(\xi) \leftarrow \mathbb{I}[S^{(j)} \leq \xi]$ for all j
    end if
    Compute $(\text{TPR}(\xi), \text{FPR}(\xi))$ over $\{(\hat{y}^{(j)}(\xi), y^{(j)})\}_{j=1}^{M}$
end for
Include ROC endpoints $(0,0)$ and $(1,1)$; sweep tied scores together
AUC $\leftarrow$ trapezoidal area along the resulting ROC curve in threshold order
return AUC

Table 4: Authoritative generation and evaluation protocol. All generations are unconditional (BOS token only), up to 512 tokens, top-k=40, top-p=0.96, following Carlini et al. (2021). A *hit* is a generation containing at least one γ-token span found verbatim in the reference index. Scenarios 1–2 use a held-out validation set of approximately 50,000 sequences and a disjoint test set; hits and rates for these rows are computed over the test set. Scenario 3 splits each model's generations 50/50 into validation and test; hits and rates for these rows are computed over the *full* generation set, so the number of positives actually evaluated in Table 2 is approximately half the value shown. Main-experiment rows populate Tables 1 and 2; the robustness protocol populates Table 10.

Scenario	**Model / configuration**	T	N **generated**	**Hits**	**Rate**
Main experiments					
1	GPT-2 XL (OpenWebText)	1.0	100,096	1,533	1.53%
2	Hubble 1B, 100B, standard	1.0	100,074	907	0.91%
2	Hubble 1B, 500B, standard	1.0	100,091	804	0.80%
2	Hubble 8B, 100B, standard	1.0	100,095	1,730	1.73%
2	Hubble 8B, 500B, standard	1.0	100,000	1,459	1.46%
2	Hubble 1B, 100B, perturbed	1.0	100,001	1,041	1.04%
2	Hubble 1B, 500B, perturbed	1.0	100,070	721	0.72%
2	Hubble 8B, 100B, perturbed	1.0	99,601	2,194	2.20%
2	Hubble 8B, 500B, perturbed	1.0	100,000	1,409	1.41%
3	apple/DCLM-7B	0.7	50,000	14	0.03%
3	TRI-ML/DCLM-1B	0.7	49,999	899	1.80%
3	Llama-3.1-8B	0.7	50,000	102	0.20%
3	Llama-3.2-1B	0.7	50,000	69	0.14%
3	Falcon-RW-1B	0.7	50,000	216	0.43%
3	Falcon-7B	0.7	49,225†	284	0.58%
3	Gemma-2-2B	0.7	40,028†	505	1.26%
3	Gemma-2-9B	0.7	35,004†	517	1.48%
3	Nemotron-H-8B	0.7	5,004†	86	1.72%
3	Pythia-410M	0.7	30,016†	270	0.90%
3	Pythia-1.4B	0.7	49,999	634	1.27%
3	Pythia-2.8B	0.7	49,998	683	1.37%
3	Pythia-6.9B	0.7	50,000	844	1.69%
3	Pythia-12B	0.7	49,999	462	0.92%
Robustness protocol (Scenario 3 only; validation = the full main-experiment run)					
3	All 14 models, disjoint test set	0.7	up to 100,000†	see Table 10	

†Generation was limited by a fixed per-model compute budget; no samples were filtered or discarded. Nemotron-H-8B generated substantially more slowly than the other models under its hybrid architecture, yielding the smallest set and correspondingly higher variance in its metrics.

C.2 Score Direction

Each method yields one scalar per sequence. Whether higher or lower values indicate a verbatim hit is fixed a priori from the method's definition (e.g. lower perplexity indicates a hit) and is not selected per configuration. We report AUC under that fixed convention, so a method whose signal is inverted in a given setting scores below 0.5 rather than being reflected above it. Tables and figures are generated from the same evaluation outputs.

C.3 GPT-2 XL Training Details

We train a GPT-2 XL architecture (1.5B parameters: 48 layers, embedding dimension 1600, 25 attention heads, context length 1024) on OpenWebText according to the recipe in the NanoGPT library (Karpathy, 2022). We use the AdamW optimizer with 120k total training steps, linear learning rate warmup from 0 to 6×10^{-4} for the first 2,000 steps and cosine annealing down to 6×10^{-5} over the remaining steps, global batch size 512.

C.4 Computational Resources

GPT-2 XL Training

Training GPT-2 XL (1.5B parameters) on OpenWebText required:

- **Hardware:** 8×NVIDIA A100 80GB GPUs
- **Duration:** 15 days of continuous training
- **Total GPU-hours:** Approximately 2,880 GPU-hours (8 GPUs × 15 days × 24 hours)
- **Software:** PyTorch 2.0, CUDA 12.1
- **Memory:** Peak GPU memory usage 75GB per GPU during training

Index Building

Exact n-gram index construction for the 500B token Hubble dataset required:

- **Hardware:** GCP c4d-highmem-384 instance (384 vCPUs, 3,024GB RAM, local NVMe SSD storage)
- **Storage:** 8.5TB for partitioned binary hash index (256 partitions)
- **Duration:** Approximately 36 hours for full 500B token indexing
- **Software:** Python 3.10, NumPy 1.24

For the 100B token dataset, index building used:

- **Hardware:** GCP high-memory instance (128 vCPUs, 256GB RAM)
- **Duration:** Approximately 16 hours

Inference and Evaluation

Sequence generation for extraction attacks and TSL score computation required:

- **Hardware:** Single NVIDIA A100 80GB GPU per model
- **Duration:**
  - Generating 100k sequences (512 tokens each): 8 hours per model
  - Computing SLD/FL scores: 4 hours per model per checkpoint
- **Memory:** Peak GPU memory 40GB for 8B models, 15GB for 1B models

Baseline Methods

Baseline method evaluations (Min-k%, Min-k%++, Neighbor, ZLIB, InfoRMIA) required:

- **Hardware:** Single A100 GPU or high-memory CPU instance depending on method
- **Duration:** 2-6 hours per baseline per model configuration

Total Compute Budget

The total computational cost for all experiments in this paper is approximately:

- **GPU compute:** 3,200 GPU-hours (primarily A100 80GB)
- **CPU compute:** 52 instance-hours on high-memory instances for index construction (≈16K vCPU-hours)
- **Storage:** 15TB for datasets, indices, and checkpoints

D Ground-Truth Construction and Localization Diagnostics

D.1 Ground-Truth Construction

Resolving matches to a single ground-truth span

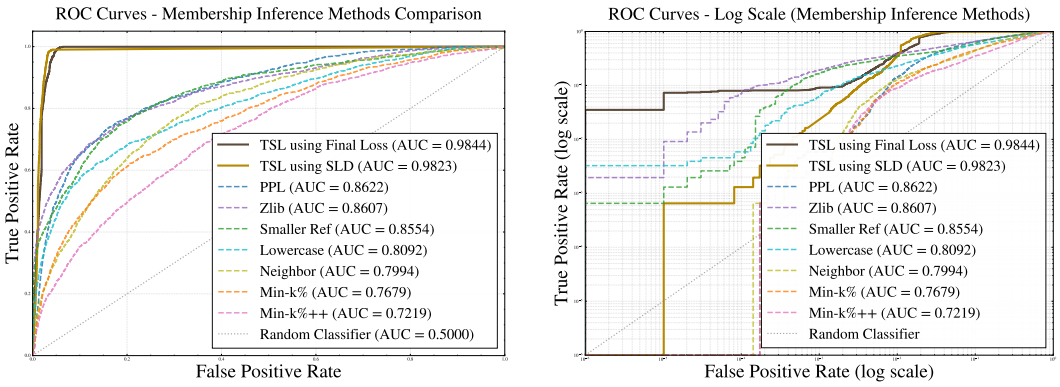
Each γ-token window hit is extended left and right while tokens continue to agree with the same source document, so a matched span is always a contiguous region of a single document rather than a chain of windows assembled across documents. We verify that $\texttt{match_len} = \gamma + \texttt{left_ext} + \texttt{right_ext}$ and $\texttt{cand_end} - \texttt{cand_start} = \texttt{match_len}$ hold for every match in our data, with no exceptions.

A generation frequently matches several source documents at the same position. This reflects duplication within the corpus rather than several distinct memorized regions: on GPT-2 XL, 85.4% of hits match more than one document, while only 8.7% contain more than one *non-overlapping* matched region (7.4% contain two, 1.2% three, 0.1% four). Where disjoint regions do occur, we take the longest single match as the ground-truth span, consistent with TSL returning its longest predicted span (Algorithm 2).

A small number of ground-truth positives admit no reference interval at all: the record flags a hit but carries no match detail, so no position in the generation is available. These account for 0.07–1.7% of hits depending on configuration and are excluded from the localization metrics, since the omission is in the reference rather than in the prediction.

D.2 ROC Curves

Figure 2 shows sequence-level verbatim-hit detection on GPT-2 XL under both linear and logarithmic axes. The log scale emphasizes the low false-positive regime, which is the operating point of interest for auditing: at deployment scale even a small false-positive rate produces a large number of outputs flagged for review that contain no indexed verbatim content. TSL with FL holds a clear advantage over the baselines throughout, and the margin is widest in that regime.



(a) Linear axes (b) Logarithmic axes

Figure 2: ROC curves for sequence-level verbatim-hit detection on GPT-2 XL generations using different memorization signals. TSL with FL outperforms all baselines, and its margin is widest in the low false-positive regime (right), which is the operating point where the cost of review is highest.

Figure 3 reports the same evaluation across the eight Hubble configurations.

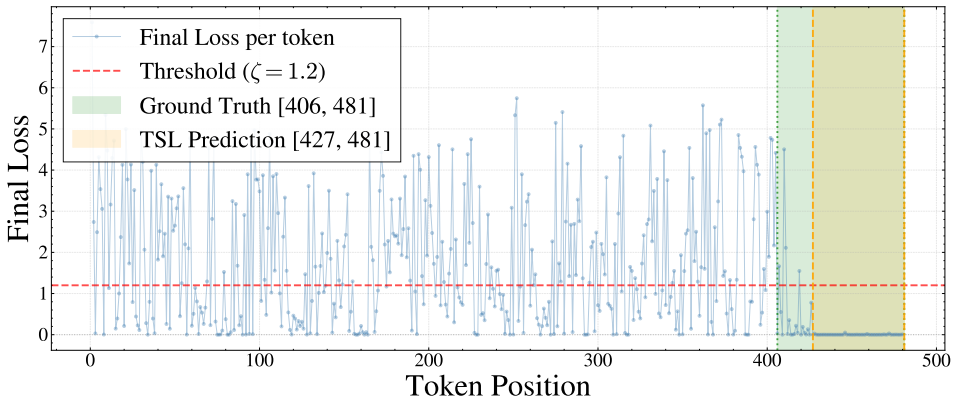
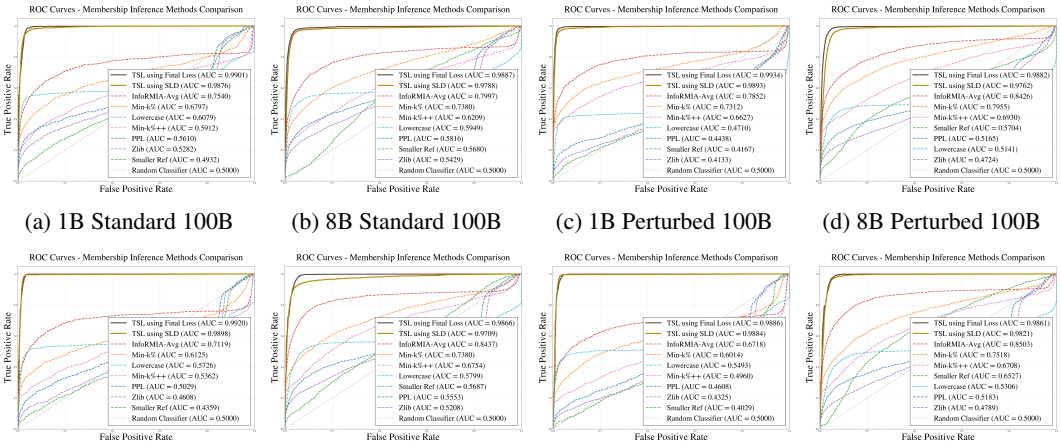


Figure 4: Visualization of TSL using FL. The red dashed horizontal line indicates the threshold ζ, while the shaded regions denote the ground-truth memorized span (green) and the span predicted by TSL (yellow).



(a) 1B Standard 100B (b) 8B Standard 100B (c) 1B Perturbed 100B (d) 8B Perturbed 100B

(e) 1B Standard 500B (f) 8B Standard 500B (g) 1B Perturbed 500B (h) 8B Perturbed 500B

Figure 3: ROC curves for sequence-level verbatim-hit detection on Hubble models. Top row: 100B tokens. Bottom row: 500B tokens. TSL with FL consistently outperforms baselines across model scales and training configurations.

D.3 TSL Visualizations

Figures 4 and 5 illustrate how memorized spans can be localized at the token level using FL and SLD respectively. A fixed threshold ζ (selected on the validation set) is used to identify contiguous low-loss token spans, and the longest span satisfying the minimum length constraint is returned as the predicted memorized region.

D.4 Hit Rates on Hubble

Scenario 2 of Table 4 reports the fraction of generated sequences containing at least one verbatim memorized γ-token span for each Hubble configuration. Hit rates range from 0.72% (500B perturbed, 1B) to 2.20% (100B perturbed, 8B). Three patterns hold across all four matched pairs: (1) 8B models show roughly twice the hit rate of 1B models at the same token budget, consistent with larger models retaining more training content; (2) checkpoints at 500B tokens exhibit lower hit rates than their 100B counterparts, indicating that continued training reduces the probability that any given generation reproduces a specific indexed sequence; (3) perturbation does not consistently change verbatim reproduction, the 8B, 100B perturbed model shows the highest rate of any configuration (2.20%), but

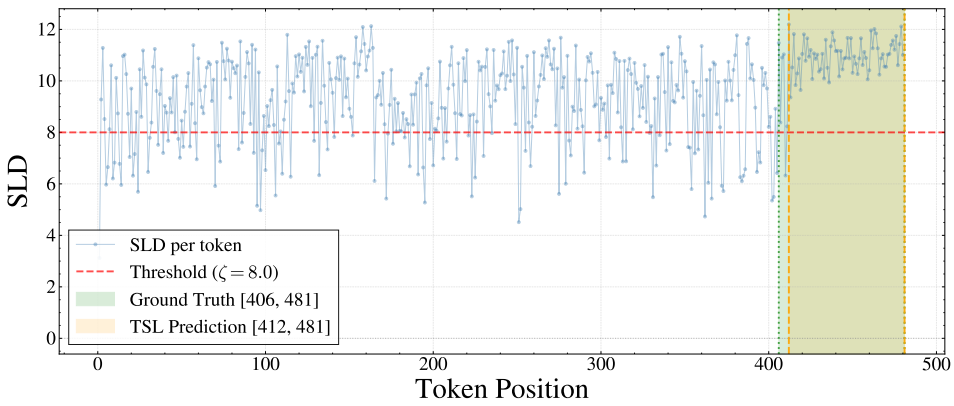


Figure 5: Visualization of TSL using SLD. The red dashed horizontal line indicates the threshold ζ, while the shaded regions denote the ground-truth memorized span (green) and the span predicted by TSL (yellow).

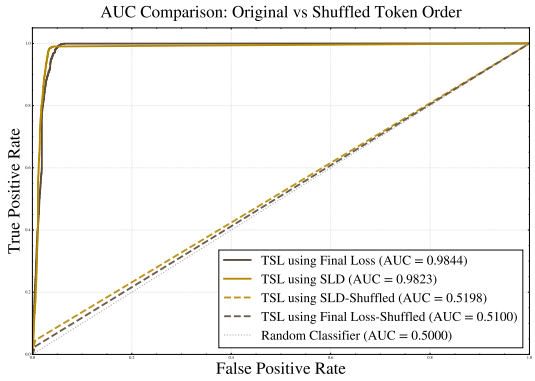


Figure 6: ROC curves for TSL using SLD and FL under the original arrangement of per-token scores and under a random permutation of those same scores. Permutation reduces AUC from ≈ 0.98 to ≈ 0.51, showing that TSL depends on the contiguity of high-scoring tokens rather than on their distribution.

the 8B, 500B perturbed model (1.41%) falls slightly below its standard counterpart (1.46%), and the 1B perturbed models are likewise mixed.

D.5 Effect of Score Order on TSL

TSL could in principle owe its performance to the *distribution* of per-token scores within a sequence rather than to their *arrangement*. We separate the two directly. For each sequence we take the per-token scores already computed by the model, permute their order uniformly at random, and re-run span extraction with the same frozen $(\zeta, L_{\min})$. The model is not re-run and the input text is not modified: the multiset of scores per sequence is identical before and after, so any change in AUC is attributable to arrangement alone.

Figure 6 reports the result. Under the original arrangement, SLD and FL reach AUCs of 0.982 and 0.984. Under permutation both collapse to ≈ 0.51, indistinguishable from a random classifier. The mechanism is visible in the span statistics: once high-scoring tokens are scattered, runs of $L_{\min}$ consecutive threshold-crossing tokens rarely survive, so extraction returns either no span or a short one whose mean is uninformative.

This isolates contiguity as the property TSL exploits. It does not, and is not intended to, characterise how the model's losses themselves would change on reordered input; that is a separate question about the model rather than about the estimator.

E Additional Counterfactual and Open-Weight Diagnostics

E.1 Additional SLD$^+$ Diagnostics

Using an early reference checkpoint

SLD is the $t_r = 0$ member of the family, taking the random initialization as its reference checkpoint. In language models this endpoint carries little document-specific information: at initialization, per-document losses are nearly constant, with standard deviation around 0.05, so the score is dominated by final loss. We therefore set t_r to an early trained checkpoint at epoch 0.2. This gives SLD$^+$ a more informative reference, where per-document loss variation has already grown to roughly 0.5 to 0.7. Table 5 shows that this change improves Spearman correlation with counterfactual memorization by 2.6 to 3.8 times across the three datasets.

Table 5: SLD (random init) vs. SLD$^+$ (early checkpoint) as memorization proxies (Spearman correlation with standard accuracy memorization, no filtering). SLD$^+$ uses the epoch-0.2 checkpoint in place of random init.

	C4 (400)	RealNews (366)	Wiki40B (369)
SLD	0.194	0.203	0.101
SLD$^+$	**0.511**	**0.527**	**0.386**
Improvement	2.6×	2.6×	3.8×

Table 6 explains why this checkpoint is useful. By epoch 0.2, the model has already completed most of its total loss reduction, 85.2% on C4, 84.8% on RealNews, and 89.4% on Wiki40B, but the per-document losses still retain substantial variation. This creates a useful contrast between the reference and final loss. Wiki40B has the lowest correlation in this group, which is consistent with the same mechanism: it is learned fastest by epoch 0.2, leaving less remaining variation for SLD$^+$ to capture.

Table 6: SLD$^+$ initial-checkpoint analysis. Standard deviation of per-document loss at each stage; "% learned" is the fraction of total loss reduction achieved by epoch 0.2.

Dataset	Init std	Ep 0.2 std	Final std	% learned	SLD$^+$ Spearman
C4	0.056	0.679	0.782	85.2%	**0.550**
RealNews	0.044	0.508	0.508	84.8%	0.538
Wiki40B	0.046	0.713	0.650	89.4%	0.393

Scaling with number of models

We check how many partition models are needed to estimate the counterfactual signal reliably (Table 7). The correlation improves rapidly at small model counts, then begins to saturate. By 200 partition models, SLD$^+$ reaches 0.502 on C4, 0.520 on RealNews, and 0.382 on Wiki40B, which is already close to the all-model estimates. The remaining gain from 200 models to the full set is small, below 0.01 Spearman on all three datasets. This suggests that the partition-model estimate is stable well before using the full set of trained models.

Counterfactual-score variants

We check whether the correlation depends on the particular counterfactual score used in the main text (Table 8). On C4, we compare the Zhang-style accuracy gap with loss-based and log-odds variants under the same filtering scheme. The accuracy-based score remains the strongest primary target, reaching 0.550 Spearman correlation after excluding the top 5% highest-simplicity documents. Loss-based memorization follows the same trend but is slightly weaker, while the log-odds variant is competitive when no simplicity filtering is applied. This check is not intended as an exhaustive sweep over all possible performance measures; rather, it verifies that the main conclusion is not specific to a single counterfactual-score definition.

Table 7: Spearman correlation vs. number of partition models (SLD^+, standard accuracy memorization). Correlation rises quickly at small N and then saturates; 200 models suffice.

N	C4	RealNews	Wiki40B
10	0.339	0.355	0.275
40	0.453	0.471	0.352
100	0.489	0.507	0.374
200	0.502	0.520	0.382
All	0.510 (399)	0.527 (366)	0.386 (369)

Table 8: C4 robustness grid across counterfactual memorization definitions and simplicity filters (400 models, SLD^+, Spearman / Pearson).

Variant	All docs	Excl top 2%	Excl top 5%	Excl top 10%
Standard (acc)	0.511 / 0.413	0.542 / 0.491	0.550 / 0.539	0.542 / 0.542
Loss-based	0.459 / 0.291	0.498 / 0.378	0.528 / 0.480	0.532 / 0.524
Log-odds	0.523 / 0.470	0.532 / 0.499	0.533 / 0.517	0.526 / 0.527

Counterfactual partition-model structure

Figure 7 shows the memorization-simplicity structure recovered by our GPT-2-small partition models. This diagnostic supports the counterfactual validation used in the main text by checking that the replicated partition-model setup exhibits the same qualitative pattern as Zhang et al. (2023).

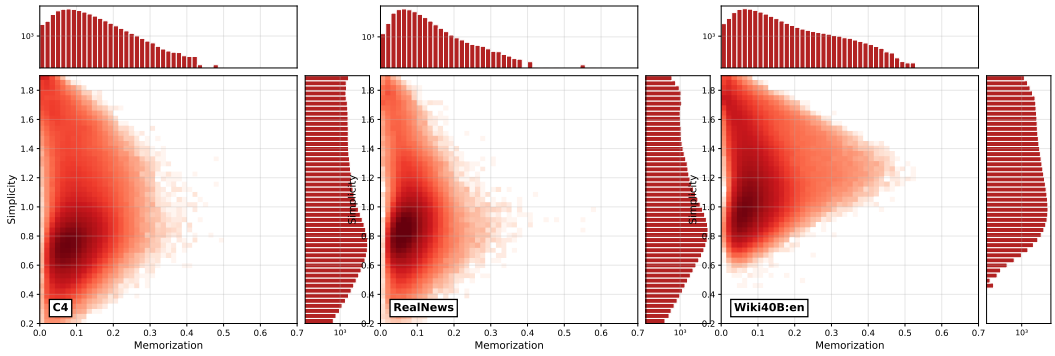


Figure 7: Replication of Zhang et al. (2023) memorization-simplicity structure using GPT-2-small.

E.2 Additional TSL and Open-Weight Diagnostics

Threshold sensitivity

TSL_{FL} requires span-extraction hyperparameters, including the threshold ζ and minimum span length $L_{\min}$. Table 9 illustrates sensitivity to ζ when $L_{\min}$ is fixed at 20: many models perform well for ζ between 2 and 5, but the best value is not universal. Some models require larger thresholds, indicating that token-loss scales differ across checkpoints. This motivates the validation protocol used in the main results, where both ζ and $L_{\min}$ are tuned on the validation split and then frozen for test evaluation.

Tokenization across model families

Each open-weight model generates in its own tokenizer. We decode each generation to text and re-tokenize it with the NeoX tokenizer, in which the DCLM index is built, so that γ-token windows are looked up in a single consistent vocabulary across all 14 models. Because the index is built in the NeoX vocabulary while each model generates in its own, ground-truth and predicted spans are established in different tokenizations for every model that is not NeoX-native. We align them by

Table 9: TSL_{FL} AUC sensitivity to threshold ζ (fixed $L_{\min} = 20$). The final column reports each model's best ζ from a sweep over $\zeta \in \{0.05, \ldots, 20\}$. The method is sensitive to ζ: optimal values vary 3–20 across models, motivating external tuning (e.g. on a validation set disjoint from evaluation).

Model	ζ=0.5	ζ=1.0	ζ=2.0	ζ=3.0	ζ=5.0	ζ=10.0	Best (ζ)
apple/DCLM-7B	0.313	0.608	0.940	0.943	0.736	0.497	**0.943** (3.0)
TRI-ML/DCLM-1B	0.441	0.723	0.898	0.929	0.677	0.675	**0.929** (3.0)
Llama-3.1-8B	0.640	0.771	0.873	0.902	0.818	0.801	**0.902** (3.0)
Llama-3.2-1B	0.628	0.758	0.871	0.925	0.913	0.850	**0.926** (4.0)
Falcon-RW-1B	0.087	0.154	0.353	0.664	0.973	0.913	**0.973** (5.0)
Falcon-7B	0.643	0.770	0.863	0.899	0.894	0.902	**0.904** (4.0)
Gemma-2-2B	0.499	0.543	0.551	0.602	0.725	0.880	**0.884** (20.0)
Gemma-2-9B	0.444	0.446	0.423	0.466	0.522	0.744	**0.841** (20.0)
Nemotron-H-8B	0.535	0.687	0.817	0.869	0.824	0.762	**0.869** (3.0)
Pythia-410M	0.642	0.767	0.871	0.920	0.921	0.825	**0.941** (4.0)
Pythia-1.4B	0.650	0.744	0.863	0.925	0.874	0.855	**0.927** (4.0)
Pythia-2.8B	0.791	0.840	0.890	0.921	0.920	0.917	**0.929** (4.0)
Pythia-6.9B	0.707	0.785	0.862	0.901	0.832	0.801	**0.901** (3.0)
Pythia-12B	0.685	0.772	0.868	0.900	0.835	0.791	**0.900** (3.0)

decoding to text and re-tokenizing, but token boundaries need not coincide exactly, so reported IoU for those models carries an additional source of error that is absent for the NeoX-native models. We therefore treat the open-weight IoU values as indicative of transfer across tokenizer families rather than as exact localization estimates.

Open-weight verbatim rates

Scenario 3 of Table 4 reports the fraction of open-weight model generations with at least one indexed γ-token verbatim span. These are indexed verbatim rates rather than total memorization rates: a positive hit indicates overlap with Hubble's 100B-token DCLM corpus, while a non-hit does not rule out memorization from another source. Rates span 0.03% (apple/DCLM-7B) to 1.80% (TRI-ML/DCLM-1B), and the models whose training corpora overlap DCLM most directly are not uniformly the highest, consistent with coverage rather than propensity driving much of the spread.

Robustness: 50K validation + 100K disjoint test

The main-paper merged table (Table 2) uses a deterministic 50/50 split of the existing generations per model (up to 50K). To verify this split is robust, we additionally evaluate under a stricter protocol: we use the full 50K generations as the validation set to tune $(\zeta, L_{\min})$ by maximizing AUC, freeze those hyperparameters, and evaluate on a disjoint, freshly generated test set of up to 100,000 sequences per model (same Carlini sampling recipe, disjoint random seed), subject to the same per-model compute budget (Table 4); per-model test-set hit counts are given in Table 10. This requires a separate matching pass against Hubble's 100B-token DCLM index. Table 10 reports the resulting metrics using the same baseline-AUC reporting convention as the main table. Across all 14 models, TSL_{FL} test AUCs are within ± 0.03 of the main-paper numbers, and TSL_{FL} again achieves the highest AUC on 12 of 14 models, with the same two Gemma-2 exceptions. Baseline AUCs and mean IoU move more, mainly on models with few hits in the main split: on apple/DCLM-7B ($N = 31$ test hits here, versus 14 hits in the entire main run) the mean-loss AUC changes from 0.631 to 0.742 and IoU from 0.625 to 0.543; mean IoU also changes on Llama-3.2-1B (0.193 to 0.349) and Nemotron-H-8B (0.606 to 0.521). For these models statistical variance dominates methodological choice.

Table 10: **Robustness check.** Localization and membership-inference performance across 14 open-weight models under a stricter split: hyperparameters $(\zeta, L_{\min})$ are tuned on the existing 50K validation set by maximizing AUC, then frozen and evaluated on a fresh disjoint test set of up to 100K sequences generated with the same Carlini sampling recipe. Mean IoU is computed only over test-set hits. Sequence-level baseline AUCs (Mean loss, Min loss, Min-$K\%$, zlib ratio) are computed per generation.

	Localization			**Membership Inference (AUC)**				
Model	Hits	IoU	Overlap (%)	Mean	Min	Min-$K\%$	Zlib	TSL
TRI-ML/DCLM-1B	1862	0.660	86.7	0.673	0.875	0.896	0.647	**0.952**
apple/DCLM-7B	31	0.543	80.6	0.742	0.808	0.923	0.870	**0.983**
Llama-3.2-1B	145	0.349	80.7	0.825	0.573	0.517	0.824	**0.947**
Llama-3.1-8B	228	0.358	82.9	0.859	0.516	0.545	0.824	**0.959**
Gemma-2-2B	1102	0.065	47.3	0.873	**0.914**	0.906	0.846	0.876
Gemma-2-9B	1425	0.067	45.5	**0.859**	0.831	0.855	0.795	0.851
Falcon-RW-1B	472	**0.675**	**94.9**	0.931	0.607	0.859	0.622	**0.963**
Falcon-7B	372	0.419	83.9	0.884	0.623	0.701	0.724	**0.888**
Nemotron-H-8B	303	0.521	84.8	0.808	0.604	0.618	0.824	**0.905**
Pythia-410M	879	0.532	88.5	0.831	0.555	0.792	0.512	**0.947**
Pythia-1.4B	1118	0.532	86.1	0.856	0.711	0.816	0.590	**0.945**
Pythia-2.8B	1401	0.433	79.4	0.914	0.815	0.918	0.580	**0.937**
Pythia-6.9B	1516	0.601	87.6	0.794	0.529	0.723	0.571	**0.900**
Pythia-12B	893	0.495	84.3	0.795	0.548	0.684	0.604	**0.903**