EXTRACTABLE MEMORIZATION FROM FIRST PRINCIPLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work on extractable memorization in language models suffers from two contrasting validity problems. Some studies overstate extraction, for example, by using sequences too short to distinguish memorization from predictability. Others imply that extraction is unreliable evidence because models can reproduce real-world text they weren't explicitly trained on. Both overlook what makes a valid extraction claim: the model must generate a training sequence with high enough probability to indicate memorization. To determine what's high enough, one has to perform a matched comparison, measuring generation probabilities of training and comparable non-training sequences. Since non-training sequences can't have been memorized, their probabilities are a baseline for predictability; exceeding this baseline is evidence of memorization. We formalize matched comparisons with (1) a conformal test calibrated to a chosen false-positive rate when sequences are sampled from populations, and (2) a single-document census that calibrates against a matched non-training document. Matched comparisons enable rigorous, calibrated memorization claims and clarify where prior setups have validity issues. On Wikipedia, OLMo 2 32B reproduces non-training 10-token suffixes roughly 24% as often as training ones: that share reflects false positives, not memorization. For Llama 3.1 70B on books, calibrated census thresholds reach as low as 10^{-27}, supporting memorization claims for sequences no feasible sampling budget would extract. We therefore refine "extractable memorization" to require both a valid memorization claim and near-certain generation within a realistic budget.

1 INTRODUCTION

For years, large language models (LLMs) have been known to leak information about their training data, sometimes reproducing exact or near-exact training sequences when prompted. The field refers to eliciting such sequences in a model's outputs as "training-data extraction," and treats extraction as an observable signal that the model "memorized" those sequences during training (Carlini et al., 2021; Lee et al., 2022). Although memorization can be studied in other ways (Shokri et al., 2017; Carlini et al., 2022), extractable memorization has drawn particular attention because of its implications for privacy and copyright (Lee et al., 2023a; Cooper & Grimmelmann, 2024; Nolte et al., 2025).

Amid a surge of research into how much LLMs memorize their training data, two recurring patterns run through recent work. On one hand, some work overstates extraction, for instance, claiming extraction for sequences too short to reliably distinguish memorization from predictability or for outputs elicited by prompts that directly leak the target sequence (Chang et al., 2023; Karamolegkou et al., 2023; Wei et al., 2025; Liu et al., 2026) On the other hand, some work argues that extraction is unreliable evidence of memorization because LLMs can also reproduce real-world text that was not included verbatim in training (Liu et al., 2025; Morris et al., 2025). Although these patterns point in opposite directions, both lose sight of what a valid extraction claim actually requires.

In this work, we revisit extractable memorization from first principles, starting with the core question: *when do generated outputs provide valid evidence of memorization in the model?* Generating a training-data sequence isn't enough on its own. To count as extraction, the model must generate it with high enough probability to indicate memorization. To determine what's "high enough," we perform a matched comparison, measuring generation probabilities of both known or suspected training-data sequences *and* comparable non-training sequences. Because non-training sequences can't have been memorized, their probabilities provide a baseline for more general predictability.

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.

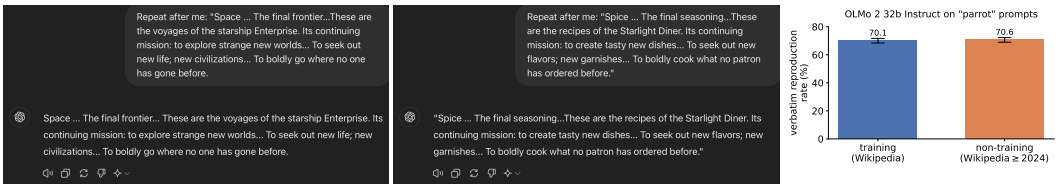


Figure 1: **Capable instruction-tuned models follow "Repeat after me" ("parrot") instructions.** (**left**) ChatGPT exactly reproduces likely training data (a *Star Trek: The Next Generation* quote), and (**middle**) non-training data (some original prose), reproduced from Cooper & Grimmelmann (2024). (**right**) We run a larger-scale version using OLMo 2 32B Instruct ($N = 3000$ each), where the matched comparison yields equal rates and carries no signal about memorization (§A.1).

Of course, not all verbatim generation is extraction. The need for a matched comparison is especially clear with an intentionally extreme example. Consider a "parrot" prompt that supplies a sequence and asks the LLM to repeat it verbatim (Fig. 1). Run only on training data, a high success rate could be mistaken for evidence of extraction. Yet capable models reproduce matched non-training sequences at essentially the same rate: they are simply copying text already in the prompt. But for our purposes, the reason the rates match is beside the point; the comparison shows that, on its own, successful verbatim generation of training data provides no basis for claiming memorization.

Asking an LLM to parrot training data is clearly not a serious extraction procedure, but the same failure arises in realistic settings. Consider a recent procedure, which searches for adversarially optimized prompts that elicit a target sequence and deems extraction successful if the resulting prompt is shorter than the target (Schwarzschild et al., 2024). The authors show success on the widely quoted line from hockey player Wayne Gretzky, "You miss 100% of the shots you don't take." We reproduce that success using Llama 2 13B, but also succeed on non-training data: a similar-length statement basketball player Jalen Brunson made in the 2026 NBA Finals, long after the training cutoff (§A.2). This does *not* imply that Llama 2 13B didn't memorize the Gretzky quotation; it simply shows that success with this procedure on training data doesn't by itself warrant a memorization claim.

When (near-)verbatim generation counts as extraction. Prior work has generally treated matched comparisons (when it runs them at all) as informal sanity checks (Carlini et al., 2023; Hayes et al., 2025b). But this is insufficient. Instead, we use the (near-)verbatim generation probabilities of matched non-training data to calibrate a threshold; a sequence's probability must clear that threshold to make a valid extraction claim, otherwise its probability could be due to more general predictability. That is, calibration makes explicit the inferential chain that prior work has left implicit:

Box 1: Making a valid extraction claim

(**a**) we measure a sequence's *generation probability* under some set of prompts; (**b**) when that probability clears the calibrated threshold, it's too high to be explained by general predictability; (**c**) this supports the conclusion that the sequence is a *member* of the training data that the model has *memorized*; (**d**) this conclusion is what licenses calling its generation probability an *extraction probability*.

Contributions. We formalize and run experiments using matched comparisons in two settings. First, for sequences sampled from populations, a conformal test calibrates memorization claims to a chosen false-positive rate (§3). On Wikipedia, for example, 24% of apparent 10-token verbatim extractions from OLMo 2 are false positives, showing that short-sequence experiments can substantially overstate memorization (§4). Second, for a fixed document where there is no population false-positive rate to control, a census instead calibrates against a matched non-training document. For Llama 3.1 70B on books, calibrated thresholds for probabilistic extraction using 50-token suffixes range from 10^{-27} to 10^{-21}. With such small thresholds, extremely low-probability sequences can support memorization claims, yet are far too improbable to sample in practice (§5). Given these results, we refine the definition of "extractable memorization" to require both a valid memorization claim from a matched comparison and near-certain generation within a realistic query budget (§6).

2 Background and Related Work

Let $\mathbb{V}$ be a token vocabulary and $\mathbb{V}^*$ the set of finite token sequences. An **autoregressive language model** θ defines a next-token distribution $q_\theta(\cdot \mid \boldsymbol{v})$ for each $\boldsymbol{v} \in \mathbb{V}^*$. Run from a **prompt** $\boldsymbol{p}$, a

decoding policy ϕ induces a distribution $q_{\theta,\phi}(\cdot \mid \boldsymbol{p})$ over **continuations**. For an ℓ-token continuation $\boldsymbol{y} = (y_1, \ldots, y_\ell)$, its generation probability is

$$q_{\theta,\phi}(\boldsymbol{y} \mid \boldsymbol{p}) \coloneqq \prod_{t=1}^{\ell} q_{\theta,\phi}(y_t \mid \boldsymbol{p}, \boldsymbol{y}_{<t}). \tag{1}$$

The **base** policy samples from the whole vocabulary ($q_{\theta,\phi} = q_\theta$), while **top-$k$** restricts at each step t to the k highest-probability tokens, renormalizes, and samples. By contrast, **greedy decoding** is deterministic, so $q_{\theta,\mathsf{greedy}}(\boldsymbol{y} \mid \boldsymbol{p}) \in \{0, 1\}$: the unique greedy continuation has probability 1 and every other continuation has probability 0.

Loosely speaking, **memorization** of a specific training sequence $\boldsymbol{x}$ means the model assigns unusually high probability to that sequence, indicating that information about $\boldsymbol{x}$ is encoded in the model. **Extraction** is a narrower phenomenon: it's the reproduction of a memorized sequence in the model's outputs via prompting. When a sequence is extractable, the model assigns it high conditional probability $q_{\theta,\phi}(\boldsymbol{x} \mid \boldsymbol{p})$, reliably reproducing $\boldsymbol{x}$ when prompted with $\boldsymbol{p}$. Not all memorized sequences are extractable (Carlini et al., 2021). A sequence can be encoded in the model yet not reliably reproduced under any prompt (Nasr et al., 2023; 2025).

Members and non-members at a chosen resolution. A model is trained on a fixed dataset $\mathbb{D}$. By definition, memorization is only possible for (near-duplicate) content present in $\mathbb{D}$: a model can't have memorized something it never saw during training. A sequence $\boldsymbol{s}$ that an extraction procedure deems to be memorized is therefore a **member** of $\mathbb{D}$. Failure to extract $\boldsymbol{s}$ is *not* evidence that it's a **non-member**, because not all training data are memorized and not all memorized training data are extractable under a given procedure. Exact-sequence membership is brittle for language models: near-duplicates are functionally members because the model has seen the content, whatever tokenization or minor syntax variation happened to accompany it (Lee et al., 2022; Maini et al., 2024). Similar to Cooper et al. (2026b), we therefore choose a resolution by fixing a candidate length ℓ, dissimilarity function $d : \mathbb{V}^\ell \times \mathbb{V}^\ell \to \mathbb{R}_{\geq 0}$, and tolerance $\varepsilon \geq 0$ (§B.1). For a candidate $\boldsymbol{s} \in \mathbb{V}^\ell$, define

$$\mathbb{B}_\varepsilon(\boldsymbol{s}) \coloneqq \{\boldsymbol{y} \in \mathbb{V}^\ell : d(\boldsymbol{y}, \boldsymbol{s}) \leq \varepsilon\}, \qquad \mathsf{membership}_\varepsilon(\boldsymbol{s}) \coloneqq \mathbf{1}\{\mathbb{D} \cap \mathbb{B}_\varepsilon(\boldsymbol{s}) \neq \varnothing\}, \tag{2}$$

where $\mathbb{D} \cap \mathbb{B}_\varepsilon(\boldsymbol{s})$ is understood as searching $\mathbb{D}$ for any length-ℓ sequence within ε of $\boldsymbol{v}$. We refer to sequences under test for memorization as **candidates** (§3). Candidate $\boldsymbol{s}$ is a near-verbatim member when $\mathsf{membership}_\varepsilon(\boldsymbol{s}) = 1$ and a non-member otherwise. Therefore, non-membership of $\boldsymbol{s}$ is deliberately stronger than requiring that $\boldsymbol{s}$ is held out from training: no $\boldsymbol{s}$ in its chosen-resolution ball $\mathbb{B}_\varepsilon(\boldsymbol{s})$ may have training support. Setting $\varepsilon = 0$ recovers verbatim (non-)membership.

Extraction procedures. Extraction involves prompting a model to attempt to reproduce a candidate $\boldsymbol{s}$, but methods that fit this loose description can differ substantially. They nevertheless share five components: a model θ, decoding policy ϕ, prompt-selection rule $f_{\mathbb{P}}$, scoring procedure score, and verification procedure verify. The rule $f_{\mathbb{P}}$ maps each candidate to a finite prompt class $\mathbb{P}(\boldsymbol{s})$, e.g., a singleton natural prefix, all natural prefixes, an adversarial search space, or another set of prompts (§B.2). For each prompt, the canonical score is the probability $q_{\theta,\phi}(\mathbb{B}_\varepsilon(\boldsymbol{s}) \mid \boldsymbol{p})$ (Eq. 1) or an approximation of it. An aggregation rule collapses per-prompt scores into the candidate score

$$\mathsf{score}(\boldsymbol{s}) = \mathsf{agg}_{\boldsymbol{p} \in f_{\mathbb{P}}(\boldsymbol{s})}\, \mathsf{score}_{\boldsymbol{p}}(\boldsymbol{s}), \tag{3}$$

where higher means more easily generated. Finally, verify uses the score either to make a memorization claim or to make no such claim.

Definition 2.1 (Extraction procedure)**.** A fixed tuple $(\theta, \phi, f_{\mathbb{P}}, \mathsf{score}, \mathsf{verify})$ defines an **extraction procedure**, where model θ and decoding policy ϕ configure the distribution $q_{\theta,\phi}$; $f_{\mathbb{P}}$ specifies the prompts used for each candidate sequence $\boldsymbol{s}$; score returns the score for $\boldsymbol{s}$ (computed under θ, ϕ, and $f_{\mathbb{P}}$); and, based on its score, verify either deems $\boldsymbol{s}$ memorized or makes no claim.

Discoverable extraction, its probabilistic and near-verbatim variants, and adversarial prompt search are all instances of Definition 2.1 (Lee et al., 2022; Carlini et al., 2023; Hayes et al., 2025b; Cooper et al., 2026a;b; Schwarzschild et al., 2024; Nasr et al., 2023; 2025) (§H.1). For our purposes, the distinction that matters is the last component: in the next section, we construct verify from matched non-members, calibrating when a generation score is high enough to support a memorization claim.

3 A CONFORMAL TEST FOR MEMORIZATION

To varying extents, prior extraction methods leave open how to define verify: the rule that decides whether a candidate's score is high enough to support a memorization claim, rather than high merely

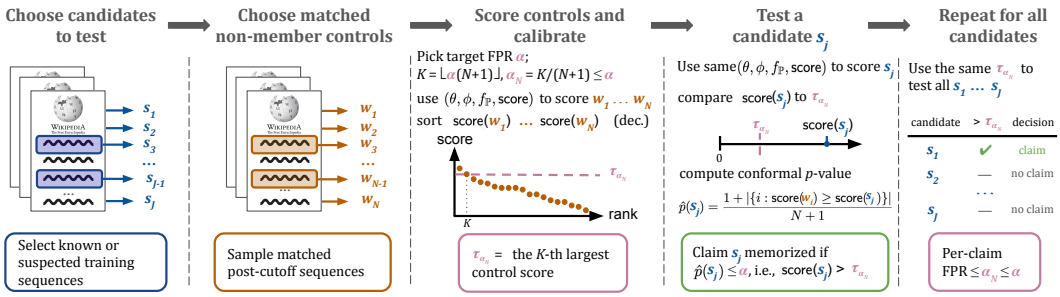


Figure 2: **Illustrating a conformal test for memorization.** (**1**) Select candidates to test (known or suspected members) and (**2**) N matched non-member controls. (**3**) Pick a target FPR α; apply the extraction score to controls to calibrate τ_{α_N}. (**4**) Apply the exact same score to a candidate and claim it's memorized only when its score exceeds τ_{α_N}. (**5**) Repeat for each candidate. If the control scores and a fresh non-member's score are exchangeable, each claim has marginal FPR at most $\alpha_N \le \alpha$.

because the continuation is predictable for a capable language model. While some work runs the same scoring procedure on non-training data and observes that reproduction is much lower (Cooper et al., 2026a; Hayes et al., 2025b), it stops short of turning this comparison into a calibrated decision.

Here, we formalize verify as a matched comparison, made precise as a **conformal test**. Recall that a candidate $\boldsymbol{s}$ is either a member or a non-member (§2). The test's null hypothesis is that $\boldsymbol{s}$ is a *matched* non-member; it rejects the null when $\mathsf{score}(\boldsymbol{s})$ exceeds a threshold calibrated from the scores of matched non-members (**controls**). Rejecting means $\boldsymbol{s}$'s generation probability is too high for it to be a matched non-member, indicating it's a member of the training data that the model memorized (Box 1). Calibration lets us make that claim at a false-positive rate (FPR) chosen in advance: how often the test claims memorization for a non-member (Fig. 2). Because the test operates on the abstract extraction procedure (Def. 2.1), it applies to any extraction method that instantiates that definition.

Matched comparisons for calibrated memorization claims. Formally, the *matched* non-member null hypothesis is $\mathsf{membership}_{\varepsilon}(\boldsymbol{s}) = 0$ (Eq. 2), with $\boldsymbol{s}$ drawn from the same distribution as the controls. **Matched** means two things. First, we select controls the same way as candidates (in practice, known or suspected members): they share the source, format, length, and other features that can affect score, except controls are known non-members. Second, the same fixed extraction procedure—model θ, decoding policy ϕ, prompt-selection rule $f_{\mathbb{P}}$, score (Eq. 3)—is applied to candidates and controls. This makes their scores exchangeable under the matched null. If the candidate were a non-member, its score would look like one more draw from the control scores' distribution.

Let $W_1, \dots, W_N \sim \mathrm{OUT}$ be the controls: N random draws from a control distribution OUT supported on non-members ($\mathsf{membership}_{\varepsilon}(W_i) = 0$). For any candidate $\boldsymbol{s}$, define

$$\widehat{p}(\boldsymbol{s}) = \frac{1 + |\{i : \mathsf{score}(W_i) \ge \mathsf{score}(\boldsymbol{s})\}|}{N+1}, \tag{4}$$

the **conformal** p**-value**. Pooling the N control scores with the candidate's, $\widehat{p}(\boldsymbol{s})$ is the fraction of those $N+1$ scores that are at least as large as $\mathsf{score}(\boldsymbol{s})$. A small value means that almost no matched non-member reaches the candidate's score. Fix a **target FPR** $\alpha \in (0,1)$ independently of the control and candidate scores. A finite control sample can only certify levels in steps of $1/(N+1)$ because the cutoff must be one of the N control scores, so write

$$K = \lfloor \alpha(N+1) \rfloor \quad \text{and} \quad \alpha_N = K/(N+1) \le \alpha. \tag{5}$$

K indexes which control score serves as that cutoff; α_N is the largest attainable **certified FPR** not exceeding α (§C.1, §C.2). The smallest nonzero level any finite control sample can certify is $1/(N+1)$, reached at $K = 1$, where the threshold is the largest control score. Because $\widehat{p} \ge 1/(N+1)$ (Eq. 4), setting $\alpha < 1/(N+1)$ gives $K = 0$ and the test can never reject (§C.1).

If the control scores in decreasing order are $\mathsf{score}_{(1)} \ge \cdots \ge \mathsf{score}_{(N)}$, then, for $K \ge 1$, the **calibrated threshold** at certified level α_N is

$$\tau_{\alpha_N} := \mathsf{score}_{(K)}, \qquad \widehat{p}(\boldsymbol{s}) \le \alpha \iff \mathsf{score}(\boldsymbol{s}) > \tau_{\alpha_N}. \tag{6}$$

The strict inequality is important: a candidate whose score is tied with the threshold doesn't pass. Because matching makes the candidate's score exchangeable with the control scores, a non-member candidate is falsely claimed as memorized with probability at most $\alpha_N \le \alpha$ (§C.2).

Proposition 1 (Finite-sample validity of the conformal test). *Fix $\alpha \in (0, 1)$ and the complete extraction procedure independently of the calibration controls and candidate scores. Let S be a fresh candidate that satisfies the null—a non-member (*$\mathsf{membership}_\varepsilon(S) = 0$*) drawn by the same rule as the controls—and suppose its score and the N control scores are exchangeable. Then*

$$\Pr_0[\widehat{p}(S) \leq \alpha] \leq \alpha_N \leq \alpha, \tag{7}$$

where $\Pr_0$ is probability under the null, taken over both the random controls and the fresh candidate, so the guarantee is marginal and the left-hand side is the FPR. For $K \geq 1$ (Eq. 5), rejection is exactly the strict threshold test in Eq. 6; for $K = 0$, the deterministic test can't reject. If the $N + 1$ scores have no ties, the first inequality is an equality; ties can only make this test conservative.

The guarantee applies per candidate: it's not conditional on the realized controls (§C.4.3), and it doesn't by itself bound the familywise error rate or the false discovery rate when many candidates are tested together (§C.5, Bates et al. (2021)). §C.2 gives the full proof, which follows the standard conformal argument (Shafer & Vovk, 2008): under exchangeability the null candidate's rank among the $N + 1$ scores is uniform, so at most K ranks reject. §C discusses the guarantee in more detail.

What the test certifies. The test enables valid, calibrated claims about memorization in LLMs that surfaces through a fixed extraction procedure (Def. 2.1), including near-verbatim memorization. At tolerance ε, clearing the threshold certifies near-verbatim membership of the candidate $\boldsymbol{s}$, not that the model memorized $\boldsymbol{s}$ exactly (Eq. 2). If $\boldsymbol{s}$ doesn't exceed the calibrated threshold (Eq. 6), no memorization claim is made (Fig. 2). This means that failing to clear the threshold is *not* evidence against memorization. The model may still encode the candidate in a way that this prompt, decoding policy, or score doesn't surface. So the test bounds false positives without saying anything about false negatives; it doesn't estimate total latent memorization in the model.

Revisiting "parrot" prompts. Recall that members and non-members are reproduced at the same rate when the LLM is prompted to "Repeat after me" (Fig. 1). In the original LLM extraction paper, Carlini et al. (2021) identified the "parrot" prompt case, and observed that no simple refinement of their definition for extraction resolved it; their solution was to disallow such prompts (§H.2). But the conformal test needs no such restriction: matched controls score just as high as candidates, the threshold calibrated from them is just as high (§4), nothing clears the threshold, and so the test makes no memorization claim. This is the only defensible outcome when members and non-members behave the same under the same extraction procedure, and isn't specific to "parrot" prompts. Any mechanism that raises scores independently of membership—copying, instruction-following, predictability, adversarial coaxing—raises the threshold, so the test also handles these cases.

What the guarantee rests on. A memorization claim is always made relative to some baseline of predictability; the conformal test just makes it explicit with the matched-non-member null, so that each claim is read against the chosen control distribution. This means the guarantee is only as good as the matched comparison behind it, and this rests on two conditions: controls must be genuine non-members and be distributionally matched to the candidates. In practice, we can only make a best effort for both (§4, §5, §7, §C.3).

4 Population rate experiments

The conformal test can avoid making invalid memorization claims for contrived setups like "parrot" prompts, but it can also do a lot more: it can support valid memorization claims about populations. To illustrate this, we estimate the population rate of calibrated verbatim memorization for Wikipedia entries, using OLMo 2 32B base. (§D reports results for 7B, 13B, and 32B.) Because the training corpus is public (OLMo et al., 2025), membership is known: the 5,000 candidate passages are members, coming from entries dated between January 2021 and the December 2023 training cutoff. We match them with 5,000 passages whose source entries are dated 2024, after the training cutoff. Each passage has a 50-token natural prefix and a suffix of either 10 or 50 tokens. We draw candidates and controls uniformly at random within their date ranges, so each is an i.i.d. sample. Note that the conformal guarantee needs only exchangeability (Prop. 1, §C.2), but in practice we additionally require independence. Independence makes the confidence intervals we plot valid (Fig. 3 & 4). It also relates α_N to the **population FPR** α^*—the fraction of the whole non-member population that the realized threshold τ_{α_N} incorrectly flags as memorized—by $\mathbb{E}[\alpha^*] = \alpha_N$ (§C.4), which our empirical estimate of calibrated memorization relies on (below, Eq. 8).

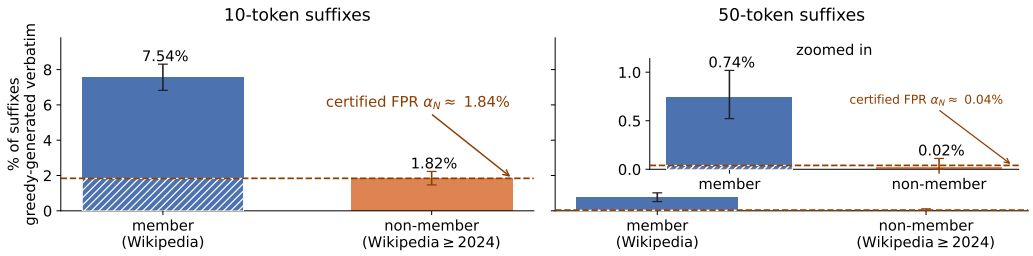


Figure 3: **Verbatim greedy generation of members and matched non-members.** For OLMo 2 32B base on Wikipedia, we run discoverable extraction for 10- (**left**) and 50-token (**right**) suffixes. Each bar is the measured rate with a 95% Clopper–Pearson interval (§D). α_N (Eq. 5) is the largest FPR the N controls can certify; it's a false-positive floor. For 10-token suffixes, $\alpha_N = 1.84\%$ against 7.54% for members; ∼24% of flagged extraction matches the rate at which the same test flags non-members. For 50-token suffixes, the floor is close to zero (see main text).

A generation rate is not a memorization rate. We begin with discoverable extraction (Lee et al., 2022; Carlini et al., 2023), the standard metric in research and model release reports (Grattafiori et al., 2024; Gemini Team et al., 2024). Because greedy decoding is deterministic, its score is binary: the continuation either exactly matches the candidate or it doesn't. There's nothing to calibrate because a binary score admits only one threshold (just one operating point). Fig. 3 contrasts verbatim generation on members with the matched non-member controls. We deliberately call it "generation" rather than "extraction": for a genuine non-member, a match can't reflect memorization.

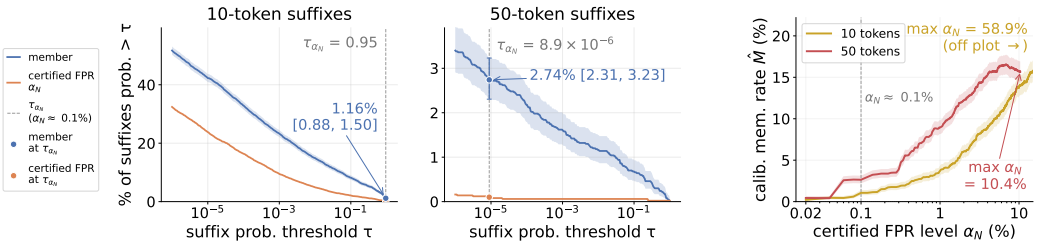
The dashed line in each panel is the **false-positive floor**: the certified FPR α_N (Eq. 5) at which the test flags sequences that can't be memorized. The bars are **in-sample rates**: for these specific 5,000-sequence samples, the fraction of members and matched controls reproduced verbatim (with 95% Clopper–Pearson intervals). For 10-token suffixes the floor is $\alpha_N = 1.84\%$ against 7.54% on members, so roughly one quarter ($1.84/7.54 \approx 24\%$) of the apparent training-data "extraction" rate is matched by the rate at which the same test flags non-members.[1] The comparison makes no claim that a particular generated member is *not* memorized. Rather, it shows that a short verbatim match is *not* a sufficiently specific membership signal: an experiment run on members alone would attribute the whole 7.54% to memorization, when the same procedure flags sequences that can't be memorized at 1.84%. At 50 tokens the picture is very different. The member generation rate is smaller (0.74%), but the non-member floor is close to zero (0.04%). That floor comes from a single false positive, which turns out to be null contamination rather than a genuine non-member (§D), so both that floor and the ≈5% share it implies are overstated.

Continuous calibration. Probabilistic extraction instead scores a continuation with $q_{\theta,\phi}(\boldsymbol{s} \mid \boldsymbol{p}) \in [0,1]$ (Eq. 1) under top-k decoding ($k = 40$), computed exactly by teacher forcing (Hayes et al., 2025b; Cooper et al., 2026a). This continuous score lets us sweep the conformal threshold across operating points, rather than expose only greedy decoding's single operating point. For each target FPR α_N, the controls set τ_{α_N} (Eq. 6) and a member is deemed memorized only if its score exceeds that threshold. Again, we see that a generation probability on its own isn't evidence of memorization. For each threshold τ, Fig. 4a plots the share of suffixes scoring above it. At 10 tokens the absolute rates are far higher, but so are the control rates (certified FPR α_N): for the smallest thresholds plotted, the member rate exceeds 50% and the certified FPR exceeds 30%; $\tau_{\alpha_N} = 0.95$, so a member must be nearly certain to clear it. At 50 tokens the member rate peaks near 3% while the controls stay near zero, and the threshold falls to $\tau_{\alpha_N} = 8.9 \times 10^{-6}$. The two raw rates aren't the right comparison: they sit at thresholds five orders of magnitude apart, each set by what the non-member controls happen to do at that suffix length. Instead, we should compare how far each member rate sits above its own control rate.

Definition 4.1 (Calibrated memorization rate)**.** For a fixed extraction procedure (Def. 2.1) and threshold τ, let $G(\tau) = \Pr[\mathsf{score}(\boldsymbol{s}) > \tau \mid \mathsf{membership}_{\varepsilon}(\boldsymbol{s}) = 1]$ and $\mathrm{FPR}(\tau) = \Pr[\mathsf{score}(\boldsymbol{s}) > \tau \mid \mathsf{membership}_{\varepsilon}(\boldsymbol{s}) = 0]$. The **calibrated memorization rate** is $M(\tau) := G(\tau) - \mathrm{FPR}(\tau)$. In practice, at the conformal operating point we estimate this by

$$\hat{M} := \hat{G} - \alpha_N, \tag{8}$$

[1] We compute the floor share from α_N rather than from the in-sample control rate because the in-sample rate is what one draw of controls happened to produce; α_N is the certified level.



(a) verbatim generation (same prefixes, 10- and 50-token suffixes) (b) calibrated memorization

Figure 4: **Generation and calibrated memorization rate for OLMo 2 32B base on Wikipedia.** (**a**) Complementary CDFs of scores, sweeping over τ: a point (τ, y) means $y\%$ of sequences score above τ (i.e., $q_{\theta,\phi}(\boldsymbol{s} \mid \boldsymbol{p}) > \tau$). 10-token suffixes have much larger absolute rates. (**b**) Calibrated memorization (Eq. 8) for 10- and 50-token suffixes across certified FPR levels α_N. The 50-token rate is the larger of the two across the sweep. (The controls contain some member contamination. For 50 tokens in particular, the plotted rate understates calibrated memorization; see §D.) Bands are 95% Clopper–Pearson intervals on the member rate, shifted by the constant α_N in (**b**).

where $\hat{G}$ is the empirical member generation rate above τ_{α_N}. Subtracting α_N is what makes $\hat{M}$ unbiased, since $\mathbb{E}[\alpha^*] = \alpha_N$ (§C.1, §C.4).

Under the matched-control assumption, the calibrated memorization rate is an extractability-based lower bound on total latent memorization in the model (§3). A suffix can be memorized yet score below τ under the fixed procedure, in which case no claim is made. Fig. 4 shows how calibration removes the short-suffix predictability floor. In contrast to the raw generation rates, the 50-token curve sits above the 10-token one across the sweep (e.g., at $\alpha_N \approx 0.1\%$, $\hat{M} = 2.64\%$ against 1.06%). Making the raw generation rate larger doesn't make the calibrated evidence for memorization stronger. Further, for memorization, the field's posture is to make conservative claims: to privilege a low FPR, even at the cost of more false negatives, because over-claiming memorization is treated as more costly than under-claiming it (Carlini et al., 2022; Cooper et al., 2026a). So while Fig. 4b plots a range of certified FPRs, the low-FPR end is where one would operate in practice (§C.4.4). (Also note that the sweep's minimum is $\alpha_N \approx 0.02\%$ because N controls certify levels only in steps of $1/(N+1)$; see §3. Certifying a smaller floor requires more controls, not a smaller α.)

5 Census Experiments

For extractable memorization, the question of interest isn't always about population rates. Sometimes we want to understand the extent to which a single specific training document is memorized. We take a **census** approach and evaluate the whole document. Unlike the Wikipedia experiment (§4), the conformal test doesn't apply; there's no population being sampled, so there is no FPR to estimate or certify. But we can still perform a matched comparison with non-training data. We do exactly this for Llama 3.1 base models, using the sliding-window verbatim probabilistic extraction procedure from Cooper et al. (2026a) (top-k decoding, $k = 40$). We test four pairs of books, where each pair matches a book in Books3 (which was included in Llama's training data) with a recent book by the same author that post-dates Llama 3.1's training cutoff. Within a pair, the books are in the same genre, and in two cases even in the same series, with overlapping characters and plot elements (§E).

The sliding-window procedure tiles each book into heavily overlapping fixed-length windows (50-token prefix, 50-token suffix $\boldsymbol{s}$), where each character is covered by many shifted suffixes (§E.2). We score every overlapping suffix given its prefix. Then, instead of a population rate, we compute whole-document **coverage** (Cooper et al., 2026a) for the member book. Let $\mathsf{span}(\boldsymbol{s})$ be the set of character indices spanned by a given 50-token $\boldsymbol{s}$. For a threshold τ and book length (in characters) L,

$$\mathsf{coverage}(\tau) = \frac{1}{L}\left|\bigcup_{\boldsymbol{s}:\mathsf{score}(\boldsymbol{s})>\tau} \mathsf{span}(\boldsymbol{s})\right|. \tag{9}$$

To pick a τ at which to compute coverage for a member book, we calibrate against its matched control book. The control's top score is how high matched non-member prose reaches, so we use it to set a pair-specific $\tau^{(\text{book})}$. By construction, no control sequence scores above it (§E.1).

Decontaminating the control to set $\tau^{(\text{book})}$. In practice, we often can't just use the control's top score to set $\tau^{(\text{book})}$. This is because a post-cutoff book may quote earlier material that's present in

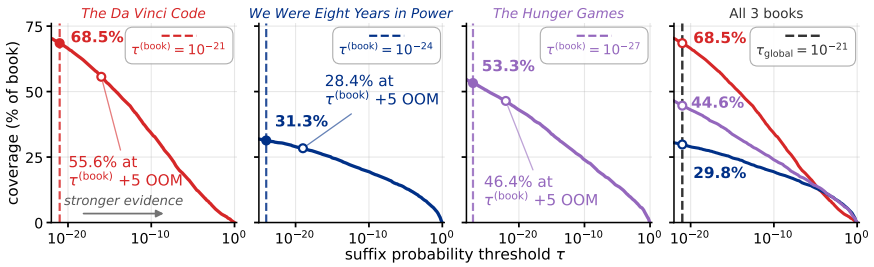


Figure 5: **Coverage as the required evidence becomes stronger.** Coverage (Eq. 9) falls as the suffix-probability threshold is raised above each matched $\tau^{(\text{book})}$. To show a common threshold across books, in the rightmost plot, we set τ_{global} to the highest (most stringent) $\tau^{(\text{book})}$, 10^{-21}, and apply it to all of the books. τ_{global} can only raise each book's bar, and so can only reduce coverage. (We don't include results for the fourth book, *Killer Instinct*, since we find effectively no evidence for memorization, as it has only one candidate window $\boldsymbol{s}$ that surpasses its $\tau^{(\text{book})} = 10^{-24}$; see §E.)

the training data. Such sequences contaminate the control pool: they're actually members, not non-members, and can score high, inflating $\tau^{(\text{book})}$. For example, *Sunrise on the Reaping* is the matched control for *The Hunger Games*, and it repeatedly quotes Edgar Allan Poe's "The Raven." We therefore remove only web-verified contaminants from the control, leave the in-training book untouched, and set $\tau^{(\text{book})}$ from the highest-scoring remaining own-prose sequence. For Llama 3.1 70B, $\tau^{(\text{book})}$ ranges from 10^{-27} to 10^{-21} (Fig. 5). These numbers are so small that they have unreliable floating-point mantissas; to be conservative, we discard the mantissa and round up to the next decade (§E.1).

Strength of evidence for memorization. The matched threshold $\tau^{(\text{book})}$ lets us make a binary claim for whether a candidate was memorized, but the strength of evidence above it clearly varies. To show this, Fig. 5 sweeps τ upward from each book's respective $\tau^{(\text{book})}$ and plots the corresponding coverage (Eq. 9). At their respective thresholds, 68.5% of *The Da Vinci Code*, 53.3% of *The Hunger Games*, and 31.3% of *We Were Eight Years in Power* are memorized. The result is robust to requiring a stronger threshold: five orders of magnitude above each $\tau^{(\text{book})}$, 55.6%, 46.4%, and 28.4% of the three books still clear their respective bars. As noted above, in absolute terms, the thresholds that give these coverage numbers are vanishingly small. They support memorization claims, but they also expose a second distinction: there are memorized sequences whose probabilities are far too small to be generated under any realistic sampling budget.

6 Realistic Extractability vs. Memorization

We make the sampling-budget point concrete by computing how many queries it would take to generate a sequence at one of these thresholds. For a fixed prompt, suppose a candidate $\boldsymbol{s}$ has generation probability $p = \mathsf{score}(\boldsymbol{s})$ on each independent attempt. After n attempts, $\boldsymbol{s}$ appears at least once with probability $\Pr[\text{at least once}] = 1 - (1-p)^n$. Consider a sequence for which $p = 10^{-21}$—the largest census threshold (§5). Generating it at least once with probability $1 - \delta = 0.95$ takes $n \approx 3 \times 10^{21}$ attempts (§F.1); at a thousand queries per second, that would take nearly $7\times$ the age of the universe. Calling this "extractable memorization" is clearly a misnomer. We therefore suggest a revised definition for **extractable memorization**:

Definition 6.1 (Extractable memorization)**.** Fix an extraction procedure (Def. 2.1), a query budget n, and a tolerance δ. A candidate $\boldsymbol{s}$ is **extractably memorized** if:

1. *it's memorized*: its score clears the matched-comparison threshold, providing calibrated evidence of memorization; and
2. *it's realistically reproducible*: a fixed procedure generates it within n queries with probability at least $1 - \delta$, whether by sampling or by another decoding procedure of comparable cost.

The two requirements answer different questions. Fig. 6 makes the distinction clear, extending the budget accounting from a single sequence to coverage of a book (Eq. 9). We compare verbatim probabilistic extraction, computed exactly by teacher forcing, against its near-verbatim variant with a maximum Levenshtein distance of 5 tokens, computed as a lower bound (Cooper et al., 2026b). Near-verbatim coverage initially dominates, but at lower probabilities the exact verbatim computation captures mass that the lower bound misses. For illustration, we take 10^5 samples as

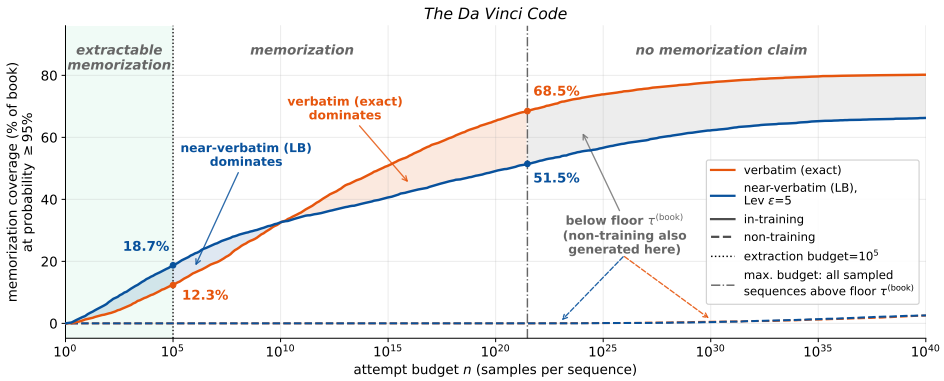


Figure 6: **Memorization and realistic extraction for *The Da Vinci Code*, Llama 3.1 70B.** Inspired by Hayes et al. (2025b), we plot book coverage (Eq. 9) with probability at least 95% within attempt budget n. Candidates reachable within 10^5 queries are extractably memorized (12.3% verbatim, 18.7% near-verbatim), and candidates above $\tau^{(\text{book})}$ but beyond that budget are memorized but not extractable (68.5% and 51.5% at $n \approx 3 \times 10^{21}$, the budget $\tau^{(\text{book})}$ implies). Candidates below $\tau^{(\text{book})}$ support no memorization claim since non-training text is also generated there.

a realistic query budget. The extraction within that budget is significantly smaller than the total memorization revealed by the matched comparison (at the budget associated with $\tau^{(\text{book})}$).

7 Conclusion

A memorization claim is only valid relative to a baseline of general predictability. We make this explicit with matched comparisons that decide when a generation probability is high enough to support a memorization claim (§2). We propose a conformal test that certifies a false-positive rate (§3) for population experiments (§4), and a census for single documents (§5)—both of which calibrate a threshold from the generation probabilities of matched non-members. Clearing that threshold licenses calling a candidate's generation probability an *extraction* probability (Box 1). But those probabilities can be vanishingly small, so we refine the definition of extractable memorization to require two things: surpassing a calibrated threshold *and* generation within a realistic query budget (§6). The consequences reach beyond our own experiments.

Perfect matched controls don't always exist (and that's okay). Calibrated claims depend on distributional matching between candidates and controls (§3). But in some cases, a perfect match isn't possible; the closest available non-members are a relatively loose match, so the null is more general. For instance, Sherlock Holmes stories are public domain and almost certainly in LLM training data; but Arthur Conan Doyle will never write a new one, so no matched non-member exists to make calibrated claims. One can still calibrate against a more general null (e.g., other non-training mysteries, a wide variety of English prose). Our census results with τ_{global} illustrate this: calibrating against general non-member prose instead of each book's own control, and taking the most conservative threshold for every book (Fig. 5). Such claims should be qualified accordingly, with clear reporting about the controls (§C.3).

Using short candidates risks validity. A short verbatim match can't carry a valid memorization claim on its own. We see this directly in our experiments: at short candidate lengths, matched non-members clear the threshold often enough to produce many false positives (§4). Yet a lot of work reports extraction using short matches without running a matched comparison (Chang et al., 2023; Karamolegkou et al., 2023; Wei et al., 2025; Liu et al., 2026). Claims in these settings rest on member generation rates, which aren't sufficient for separating memorization from general predictability.

Extraction is evidence for memorization, when it's calibrated. While that work risks overclaiming, other recent work claims the opposite: that extraction is unreliable evidence of memorization, because models can verbatim complete text they weren't explicitly trained on (Liu et al., 2025; Morris et al., 2025). But this work doesn't account for near-duplicate members contaminating the sequences they treat as non-members (§C.3), and, regardless of this issue, that conclusion is too strong. Valid extraction claims are always relative to a specific extraction procedure and baseline, so a negative result for one procedure doesn't say anything about calibrated claims under another.

AI USE STATEMENT

In this work, we used generative AI tools for identifying typos in the manuscript, soliciting feedback on plots concerning accessibility for readers with visual impairments, for a collaborative (human-led) agent skill we developed for decontaminating nulls (see §E), and checking for duplicated or overloaded symbols across the mathematical notation.

We haven't used generative AI tools for the following tasks that require disclosure: generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. With the exception of the agent skill mentioned above, we have not used generative AI tools to implement methods.

We haven't used generative AI for the following tasks with recommended disclosure: formulate questions for surveys or interviews, create or modify scientific figures or images, suggest experimental parameters, create or edit software code, creation of artifacts, draft parts of a research paper, transcribe recordings of research material, summarize or analyze existing literature, discover research topics or identify gaps, brainstorming, sourcing/searching for information, identify relevant literature, format references, suggest a structure for a research paper, propose a title or keywords for a research paper.

We have reviewed all AI-assisted work. For example, for typo checking, we created a git diff between the human-written paper and the AI output, and manually verified each change before implementing it in the draft. The agent skill we developed for decontaminating nulls is iterative and requires human inspection and verification at every single step.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Scientific papers on memorization and extraction are now frequently cited in litigation. Claims that cite this literature don't always engage with whether the specific evidence follows a valid extraction procedure, or with whether the claim being made overstates what that evidence can support (§1, §7). Our work speaks directly to this issue by revisiting extractable memorization from first principles, similar to how Carlini et al. (2022) revisited "Membership Inference Attacks From First Principles." We formalize what makes the generation of (near-)verbatim training data valid empirical evidence for a claim of extraction, and therefore of memorization in the model. We hope this work will help courts evaluating expert testimony, with respect to assessing the scientific validity of the methods used to claim extraction of memorized training data.

REPRODUCIBILITY STATEMENT

We release the code for running the experiments in §4, §5, §D, §E, and §F at [redacted]. We can't distribute the books for which we ran the census experiments, as they're under copyright. Instead, we provide links to public domain books in `.txt` format that can be used with our code base. For the theoretical results, we complement the brief discussion of assumptions in §3 with further details in §C, where we provide complete proofs. We summarize the assumptions behind all of our results in §G.

REFERENCES

Ahmed Ahmed, A. Feder Cooper, Sanmi Koyejo, and Percy Liang. Extracting books from production language models. *arXiv preprint arXiv:2601.02671*, 2026.

Federico Barbero, Xiangming Gu, Christopher A. Choquette-Choo, Chawin Sitawarin, Matthew Jagielski, Itay Yona, Petar Veličković, Ilia Shumailov, and Jamie Hayes. Extracting alignment data in open models, 2025. URL https://arxiv.org/abs/2510.18554.

Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *arXiv preprint arXiv:2104.08279*, 2021.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29(4):1165–1188, 2001.

Dan Brown. *The Da Vinci Code*. Transworld Publishers; Bantam Press; Corgi, 2003.

Dan Brown. *The Secret of Secrets*. Doubleday; Penguin Random House LLC, 2025.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pp. 267–284, 2019.

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, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying Memorization Across Neural Language Models. In *International Conference on Learning Representations*, 2023.

Kent K. Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. Speak, Memory: An Archaeology of Books Known to ChatGPT/GPT-4. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7312–7327, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.453. URL https://aclanthology.org/2023.emnlp-main.453/.

Ta-Nehisi Coates. *We Were Eight Years in Power*. One World; Random House; Penguin Random House LLC, 2017.

Ta-Nehisi Coates. *The Message*. One World; Random House; Penguin Random House LLC, 2024.

Suzanne Collins. *The Hunger Games*. Scholastic Children's Books; Scholastic Ltd, 2008.

Suzanne Collins. *Sunrise on the Reaping*. Scholastic Press; Scholastic, 2025.

A. Feder Cooper and James Grimmelmann. The Files are in the Computer: On Copyright, Memorization, and Generative AI. *arXiv preprint arXiv:2404.12590*, 2024.

A. Feder Cooper, Mark A. Lemley, Allison Casasola, Ahmed Ahmed, Aaron Gokaslan, Amy B. Cyphert, Christopher De Sa, Daniel E. Ho, and Percy Liang. Extracting memorized pieces of (copyrighted) books from open-weight language models. In *Third Conference on Language Modeling*, 2026a. URL https://openreview.net/forum?id=VE3f9mjaCd.

A. Feder Cooper, Mark A. Lemley, Christopher De Sa, Lea Duesterwald, Allison Casasola, Jamie Hayes, Katherine Lee, Daniel E. Ho, and Percy Liang. Estimating near-verbatim extraction risk in language models with decoding-constrained beam search. In *Third Conference on Language Modeling*, 2026b. URL https://openreview.net/forum?id=05XN0gq1Z7.

Joseph Finder. *Killer Instinct*. Orion Publishing Group, 2006.

Joseph Finder. *The Oligarch's Daughter*. Harper; HarperCollins, 2025.

Gemini Team, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Aaron Grattafiori et al. The Llama 3 Herd of Models, 2024. URL https://arxiv.org/abs/2407.21783.

Jamie Hayes, Ilia Shumailov, Christopher A. Choquette-Choo, Matthew Jagielski, Georgios Kaissis, Milad Nasr, Meenatchi Sundaram Muthu Selva Annamalai, Niloofar Mireshghallah, Igor Shilov, Matthieu Meeus, Yves-Alexandre de Montjoye, Katherine Lee, Franziska Boenisch, Adam Dziedzic, and A. Feder Cooper. Exploring the limits of strong membership inference attacks on large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=x0i7wvRLHK.

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 Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *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, Albuquerque, New Mexico, April 2025b. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.469/.

John Hughes, Sara Price, Aengus Lynch, Rylan Schaeffer, Fazl Barez, Sanmi Koyejo, Henry Sleight, Erik Jones, Ethan Perez, and Mrinank Sharma. Best-of-N Jailbreaking, 2024. URL https://arxiv.org/abs/2412.03556.

Daphne Ippolito, Florian Tramer, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher Choquette Choo, and Nicholas Carlini. Preventing Generation of Verbatim Memorization in Language Models Gives a False Sense of Privacy. In C. Maria Keet, Hung-Yi Lee, and Sina Zarrieß (eds.), *Proceedings of the 16th International Natural Language Generation Conference*, pp. 28–53, Prague, Czechia, September 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.inlg-main.3. URL https://aclanthology.org/2023.inlg-main.3/.

Najeeb Jebreel, Mona Khalil, David Sánchez, and Josep Domingo-Ferrer. Revisiting the LiRA Membership Inference Attack Under Realistic Assumptions, 2026. URL https://arxiv.org/abs/2603.07567.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright Violations and Large Language Models, 2023. URL https://arxiv.org/abs/2310.13771.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating Training Data Makes Language Models Better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, pp. 8424–8445, 2022.

Katherine Lee, A. Feder Cooper, and James Grimmelmann. Talkin' 'Bout AI Generation: Copyright and the Generative-AI Supply Chain. *arXiv preprint arXiv:2309.08133*, 2023a.

Katherine Lee, A. Feder Cooper, James Grimmelmann, and Daphne Ippolito. AI and Law: The Next Generation. *SSRN*, 2023b. http://dx.doi.org/10.2139/ssrn.4580739.

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, 2025. URL https://arxiv.org/abs/2503.17514.

Xinyue Liu, Niloofar Mireshghallah, Jane C. Ginsburg, and Tuhin Chakrabarty. Alignment Whack-a-Mole : Finetuning Activates Verbatim Recall of Copyrighted Books in Large Language Models, 2026.

Bruce T. Lowerre. *The HARPY Speech Recognition System*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1976. PhD Dissertation.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. LLM Dataset Inference: Did you train on my dataset?, 2024. URL https://arxiv.org/abs/2406.06443.

John X. Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G. Edward Suh, Alexander M. Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. How much do language models memorize?, 2025. URL https://arxiv.org/abs/2505.24832.

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.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable Extraction of Training Data from Aligned, Production Language Models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vjel3nWP2a.

Gonzalo Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, March 2001. ISSN 0360-0300. doi: 10.1145/375360.375365. URL https://doi.org/10.1145/375360.375365.

Robert G. Newcombe. Improved confidence intervals for the difference between binomial proportions based on paired data. *Statistics in Medicine*, 17(22):2635–2650, 1998a.

Robert G. Newcombe. Interval estimation for the difference between independent proportions: comparison of eleven methods. *Statistics in Medicine*, 17(8):873–890, 1998b.

Henrik Nolte, Michèle Finck, and Kristof Meding. Machine learners should acknowledge the legal implications of large language models as personal data, 2025. URL https://arxiv.org/abs/2503.01630.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 Furious, 2025. URL https://arxiv.org/abs/2501.00656.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pp. 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

Aneesh Pappu, Billy Porter, Ilia Shumailov, and Jamie Hayes. Measuring memorization in RLHF for code completion. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Tg8RLxpMDu.

Hung Viet Pham, Shangshu Qian, Jiannan Wang, Thibaud Lutellier, Jonathan Rosenthal, Lin Tan, Yaoliang Yu, and Nachiappan Nagappan. Problems and opportunities in training deep learning software systems: An analysis of variance. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2020.

Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C Lipton, and J Zico Kolter. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9, 2008.

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.

Vladimir Vovk. Conditional validity of inductive conformal predictors. *arXiv preprint arXiv:1209.2673*, 2012.

Johnny Tian-Zheng Wei, Maggie Wang, Ameya Godbole, Jonathan H. Choi, and Robin Jia. Interrogating llm design under a fair learning doctrine, 2025. URL https://arxiv.org/abs/2502.16290.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.

Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 58244–58282. PMLR, 2024.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual Memorization in Neural Language Models, 2023. URL https://arxiv.org/abs/2112.12938.

Donglin Zhuang, Xingyao Zhang, Shuaiwen Leon Song, and Sara Hooker. Randomness in neural network training: Characterizing the impact of tooling. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 4, 2022.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and Transferable Adversarial Attacks on Aligned Language Models, 2023. URL https://arxiv.org/abs/2307.15043.

APPENDIX TABLE OF CONTENTS

A Motivating examples **16**
A.1 "Parrot" prompts experiment . . . 16
A.2 Generating verbatim non-training data with ACR . . . 16

B Extended preliminaries **17**
B.1 Members, non-members, and the membership indicator . . . 18
B.2 Defining an extraction procedure . . . 19

C Extended details on the conformal test for memorization **20**
C.1 Notation and summary of relevant quantities . . . 20
C.2 Finite-sample validity of the conformal test . . . 23
C.3 Practical limits of matching . . . 25
C.4 The population false-positive rate of the threshold . . . 26
C.4.1 The Beta law . . . 27
C.4.2 Our score isn't always continuous . . . 28
C.4.3 Conditional guarantees . . . 32
C.4.4 The point of all this in practice . . . 33
C.5 Multiple testing across candidates . . . 33

D Population experiments **36**
D.1 Plotted intervals . . . 36
D.2 Score distributions and contamination . . . 37
D.3 Varying suffix length . . . 37
D.4 Testing a different decoding policy . . . 40
D.5 Margin . . . 41

E Census experiments **43**
E.1 Calibrating thresholds and decontamination . . . 43
E.2 Additional results . . . 46

F Extractability at a finite query budget **47**
F.1 Query budget . . . 47
F.2 Population budget views . . . 47
F.3 Census budget views . . . 49
F.4 Deterministic recovery . . . 50

G Assumptions used throughout the paper **51**

H Extended discussion of prior work **53**
H.1 Prior work on extraction, in these terms . . . 53
H.2 Prior work in light of matched comparisons . . . 53
H.3 Validity in prior work . . . 55
H.4 Membership inference . . . 56

A MOTIVATING EXAMPLES

In this appendix, we provide details on the experiments discussed in §1 concerning "parrot" prompts (§A.1) and the extraction procedure of Schwarzschild et al. (2024) (§A.2).

A.1 "PARROT" PROMPTS EXPERIMENT

The experiment in Fig. 1 involves OLMo 2 32B Instruct, and we use the following prompt template:

```
Repeat the text after the newline exactly, word for word.  Do not
continue, summarize, or add anything; output only that text:\n{X}
```

where `{X}` is the first 50 tokens of either a Wikipedia entry in OLMo 2's training data (a member) or a post-cutoff Wikipedia entry (a non-member). We run this prompt template for 3,000 member sequences and 3,000 non-member sequences. For each prompt, we greedy generate 55 tokens—slightly more than 50, so that the decoded text is at least as long as `{X}` even when the tokenization differs—decode to text, truncate the decoded result to `{X}`'s character length, and require a character-exact match to `{X}`. (It's easiest to do this, rather than comparing in token space, since the inclusion of the prompt template with `{X}` may result in the portion containing `{X}` having a different tokenization than the generation of `{X}`, even if the underlying text is identical.)

Each scored sequence—here, with greedy decoding and checking an exact match—is an independent success/failure trial, so each reported rate is a binomial proportion. In Fig. 1, we therefore show 95% Clopper–Pearson intervals. Intuitively, the interval reflects how much the measured rate could shift had we drawn a different 3,000 sequences. The member rate is 70.1% (95% CP interval $[68.4, 71.7]$) and the non-member rate is 70.6% ($[69.0, 72.3]$). The two intervals overlap, so the member and non-member rates are statistically indistinguishable.

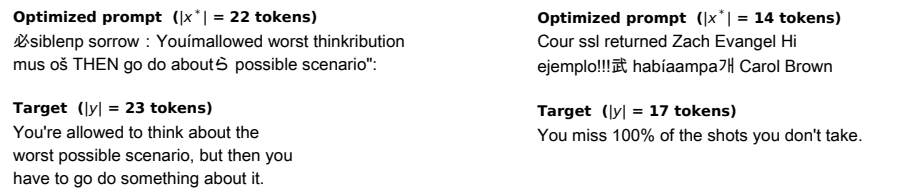
Pretend that one assumes this is a valid extraction procedure (even though it clearly isn't). Because it behaves the same on members and non-members, the matched comparison on its own shows that successful generation provides no basis for claiming memorization.

A.2 GENERATING VERBATIM NON-TRAINING DATA WITH ACR

The same issue can arise in a less contrived procedure. Schwarzschild et al. (2024) search for an adversarially optimized prompt that elicits a target sequence and deem extraction successful when that prompt is shorter than the target. To measure this, they introduce **adversarial compression ratio (ACR)**, which divides the length (in tokens) of the target by that of the prompt; extraction is successful if that ratio exceeds 1. We were able to achieve success for both training and non-training data; in each case, however, we had to make some changes to the released code, which we discuss below.

They define success by greedy decoding, but the code implementation tests it by teacher-forcing the $\arg\max$ within GCG (Zou et al., 2023). GCG searches for a prompt by minimizing the mean per-token cross-entropy of the target; the implementation stops as soon as the $\arg\max$ agrees with the target at every position. In arithmetic the two criteria coincide: if the $\arg\max$ is the target token at each step, greedy decoding stays aligned with the target throughout generation. However, teacher forcing and greedy decoding can exhibit slight numeric differences. (The former uses a single forward pass over the whole sequence with no KV cache; the latter does incremental generation with a KV cache.) Because the search halts at the first exact match, we observed that its solutions often sit on a very thin $\arg\max$ margin, where a small numerical difference can flip one token and cause the continuation to diverge from that point on. (This $\arg\max$ criterion is fragile by construction.)

In practice, we found that prompts the released code marks as successful (via $\arg\max$ teacher forcing) often fail to reproduce the target verbatim under greedy generation. The released code does call `generate` once a prompt succeeds, but only logs the result: it's never compared against the target, and never feeds back into the success flag, the reported ACR, or the saved results. (This is how we observed the target-generation failures to begin with.) Across 16 runs (8 targets on each of Llama 2 13B and Llama 2 13B Chat), we observed 11 such failures: prompts marked as successful that don't greedily generate the target verbatim, and therefore fail the paper's own definition. (We verified this against the released code on our own runs; we don't claim that the published numbers reflect this error, since we don't know what the underlying results were.)



(a) Post-cutoff (Brunson).

(b) Pre-cutoff (Gretzky).

Figure 7: **Optimized prompts that satisfy the ACR criterion at $b = 0.9$.** Each panel shows the GCG-optimized prompt above its target, with token counts. Both prompts are shorter than their targets, so both clear $\mathrm{ACR} > 1$ under Schwarzschild et al. (2024)'s criterion, and both elicit the target with geometric-mean per-token probability above 0.9 using Llama 2 13B. Note that ACR doesn't place any constraints on prompt/target overlap. The Brunson prompt reuses target content words; the Gretzky prompt doesn't.

We therefore relaxed the success condition rather than the definition. We keep teacher forcing, but instead of requiring the $\arg\max$ to match at every position, we require the target's teacher-forced probability to be high: success when the geometric-mean per-token probability of the target is at least b. (This is analogous to doing probabilistic extraction (Hayes et al., 2025b) instead of discoverable extraction (Carlini et al., 2023), but in the context of GCG.) This is equivalent to a bound on GCG's own objective, $\mathrm{loss} \leq -\ln b$, and so doesn't depend on target length. We set $b = 0.9$, so that a successful prompt is one from which the target would be reproduced with high probability under sampling. The length search is the same, so the reported ratio is still $\mathrm{ACR} = |y|/|x^*|$, where y is the target and x^* is the optimized prompt.

With this relaxed success condition, we applied the procedure to the Wayne Gretzky line, "You miss 100% of the shots you don't take" (one of the shipped targets in the repository), and we achieved success using Llama 2 13B: a 14-token prompt for the 17-token target, $\mathrm{ACR} = 1.21$ (Fig. 7b). But we also succeeded for a similar-length statement Jalen Brunson made following the Knicks' 29-point comeback against the Spurs in the 2026 NBA Finals: "You're allowed to think about the worst possible scenario, but then you have to go do something about it"—a 22-token prompt for the 23-token target, $\mathrm{ACR} = 1.05$ (Fig. 7a). The latter post-dates the model's training cutoff. This doesn't show that the model failed to memorize the Gretzky quotation; it shows that satisfying the procedure's criterion doesn't, by itself, warrant a memorization claim.

We also observed that the optimized prompts often contain words from the target itself. This is visible even in Schwarzschild et al. (2024)'s own teaser figure: for the 12-token target "To be or not to be, that is the question.", the 4-token prompt they report is `urgesTOBE quote!`, which hands the model the target's opening words (their Fig. 1). The Brunson prompt is a far more pronounced instance of the same thing (Fig. 7a).

The point is, there is nothing in GCG itself that prevents leaking the target in the prompt. One could address this by changing the definition of ACR by adding a penalization term for overlap: replace $|x^*|$ in the denominator with $|x^*| + \lambda o$, where o counts the number of target tokens (or some other overlap count) the prompt reuses and $\lambda \geq 0$ sets the cost. Setting $\lambda = 0$ recovers the original ACR definition. Note that doing this would affect the results on Brunson, but not Gretzky; depending on the overlap count and penalty term, it could push the Brunson ACR below 1 (i.e., it would no longer be a success). (We raise this as an observation, rather than a metric we've validated.)

B Extended preliminaries

In this appendix, we give the preliminaries in full. Let $\mathbb{V}$ be a token vocabulary and $\mathbb{V}^*$ the set of finite token sequences. An **autoregressive language model** θ defines a next-token distribution $q_\theta(\cdot \mid \boldsymbol{v})$ for each $\boldsymbol{v} \in \mathbb{V}^*$. Run autoregressively from a **prompt** $\boldsymbol{p}$, a **decoding policy** ϕ induces a distribution $q_{\theta,\phi}(\cdot \mid \boldsymbol{p})$ over **continuations**. The **base** policy makes no adjustment ($q_{\theta,\phi} = q_\theta$): it samples from the whole vocabulary at each step. **Top-k decoding** restricts at each step to the k highest-probability tokens, renormalizes, and samples. By contrast, **greedy decoding** is deterministic: it selects the highest-probability ($\arg\max$) token at each step. Each policy's next-token distribution depends only on the preceding tokens, so it factorizes autoregressively over continuations.

For a prompt $\boldsymbol{p}$ and an ℓ-length continuation $\boldsymbol{y} = (y_1, \ldots, y_\ell) \in \mathbb{V}^\ell$, the generation probability is

$$q_{\theta,\phi}(\boldsymbol{y} \mid \boldsymbol{p}) \coloneqq \prod_{t=1}^{\ell} q_{\theta,\phi}(y_t \mid \boldsymbol{p}, \boldsymbol{y}_{<t}). \tag{10}$$

For greedy decoding ($\phi = \mathsf{greedy}$), $q_{\theta,\mathsf{greedy}}(\boldsymbol{y} \mid \boldsymbol{p}) \in \{0, 1\}$: the unique, deterministic ℓ-token continuation has probability 1, and all other ℓ-token continuations have probability 0.

Loosely speaking, **memorization** of a specific training sequence $\boldsymbol{x}$ means that the model assigns unusually high probability to that sequence, indicating that information about $\boldsymbol{x}$ is encoded in the model. **Extraction** is a narrower phenomenon: it is the reproduction of a memorized sequence in the model's outputs via prompting. When a sequence is extractable, the model assigns it high conditional probability $q_{\theta,\phi}(\boldsymbol{x} \mid \boldsymbol{p})$, and reliably reproduces $\boldsymbol{x}$ when prompted with $\boldsymbol{p}$. Not all memorized sequences are extractable (Carlini et al., 2021). A sequence can be encoded in the model yet not reliably reproduced under any prompt (Nasr et al., 2023; 2025). And studying memorization does not require prompting in general; other approaches, such as membership inference, study memorization using different methods (Shokri et al., 2017; Carlini et al., 2019; 2022; Hayes et al., 2025a).

Memorization is only possible for data the model was trained on, so we need to make clear what it means for a sequence to be a "member" of the training data, and what it means not to be (§B.1). We then pin down a definition for an extraction procedure, which formalizes the intuition above (§B.2).

B.1 Members, non-members, and the membership indicator

A language model is trained on a **training dataset** $\mathbb{D}$, which is a fixed collection of token sequences. We make no commitment to the structure or format of $\mathbb{D}$: it may be a set of documents of varying or fixed lengths, or have any other organization amenable to being ingested by a training procedure. By definition, memorization is only possible for content present in $\mathbb{D}$: a model cannot have memorized something it never saw during training. A sequence $\boldsymbol{s}$ that an extraction procedure deems to be memorized is therefore necessarily a **member** of the model's training data. Failure to extract $\boldsymbol{s}$ is *not* evidence that $\boldsymbol{s}$ is a **non-member**: not all training data are memorized, and not all memorized training data are necessarily extractable under a given procedure. Throughout, we refer to sequences $\boldsymbol{s}$ as **candidates** under test for memorization.

Including near-duplicates when defining membership. Near-duplicates of training sequences are functionally members of the training data: the model has seen the content, whatever tokenization or minor syntax variation happened to accompany it (Lee et al., 2022). As a result, restricting a definition of training-data membership to exact token sequences is brittle for language models (Zhang et al., 2023; Maini et al., 2024; Hayes et al., 2025a): it would misclassify near-duplicates of members as non-members. We therefore define membership at the level of near-verbatim variations of a sequence, rather than at the granularity of the sequence itself.

To make "near-verbatim" precise, fix a dissimilarity function $d : \mathbb{V}^\ell \times \mathbb{V}^\ell \to \mathbb{R}_{\geq 0}$ that measures how different two length-ℓ sequences are, such as token-level edit distance (Navarro, 2001) or character-level BLEU score (Papineni et al., 2002).[2] For tolerance $\varepsilon \geq 0$, the **near-verbatim ε-ball** around a length-ℓ sequence $\boldsymbol{v}$ is

$$\mathbb{B}_\varepsilon(\boldsymbol{v}) \coloneqq \{\boldsymbol{y} \in \mathbb{V}^\ell : d(\boldsymbol{y}, \boldsymbol{v}) \leq \varepsilon\}, \tag{11}$$

the length-ℓ sequences within ε of $\boldsymbol{v}$.[3] Setting $\varepsilon = 0$ recovers the singleton, verbatim $\mathbb{B}_0(\boldsymbol{v}) = \{\boldsymbol{v}\}$.[4]

Definition B.1 (Membership indicator)**.** An arbitrary length-ℓ sequence $\boldsymbol{v}$ is a near-verbatim training-data member at tolerance ε if some length-ℓ sequence in $\mathbb{D}$ is within distance ε of $\boldsymbol{v}$; otherwise it is a non-member. The corresponding **membership indicator** is

$$\mathsf{membership}_\varepsilon(\boldsymbol{v}) \coloneqq \mathbf{1}\{\mathbb{D} \cap \mathbb{B}_\varepsilon(\boldsymbol{v}) \neq \varnothing\}, \tag{12}$$

where $\mathbb{D} \cap \mathbb{B}_\varepsilon(\boldsymbol{v})$ is understood as the result of searching $\mathbb{D}$ for any length-ℓ sequence within ε of $\boldsymbol{v}$. When $\mathbb{D}$ contains documents of varying lengths, this amounts to checking length-ℓ subsequences.

[2] While not strictly necessary, for convenience we restrict to length-ℓ sequences. This matches how extraction procedures typically compare candidate sequences and generated sequences of common length (Cooper et al., 2026b).

[3] We omit d from the notation; once set, we consider it fixed, while different analyses may vary ε.

[4] Memorization research tends to restrict to such simple dissimilarity functions, applied with low tolerances ε, in part because it is hard to design a semantically sound similarity function for natural text that captures memorization signal without conflating it with generalization (Barbero et al., 2025).

When $\mathsf{membership}_\varepsilon(\boldsymbol{v}) = 1$, $\mathbb{B}_\varepsilon(\boldsymbol{v})$ has training support: at least one training sequence falls within distance ε of $\boldsymbol{v}$. Conversely, when $\mathsf{membership}_\varepsilon(\boldsymbol{v}) = 0$, no sequence in the ε-ball $\mathbb{B}_\varepsilon(\boldsymbol{v})$ was present in training. One practical consequence is that non-membership of $\boldsymbol{v}$ is deliberately stronger than simply requiring that $\boldsymbol{v}$ is held-out from training. A candidate $\boldsymbol{s}$ whose ε-ball contains a training near-duplicate is *not* a valid non-member, regardless of whether $\boldsymbol{s}$ itself appears in the training data.

B.2 Defining an extraction procedure

Extraction involves prompting a model to attempt to reproduce a candidate sequence, where the candidate is either known or suspected to be a member of the training data. There are many approaches in the literature that satisfy this loose definition, but they all have the same five components. The first two are the model θ and the decoding policy ϕ. The remaining three are a prompt-selection rule that picks the prompts the model and decoding policy are tested against, a scoring procedure that evaluates how easily the model and decoding policy reproduce the candidate under those prompts, and a verification procedure that judges whether the candidate is memorized based on its score.

Prompt-selection rule. For a candidate $\boldsymbol{s}$, a **prompt class** for $\boldsymbol{s}$ is a finite set of prompts $\mathbb{P}(\boldsymbol{s}) \subseteq \mathbb{V}^*$. A prompt class can be a singleton (e.g., just the 50-token natural prefix of a training sequence), or a larger set. For instance, the prompt class can be extended to all natural prefixes of $\boldsymbol{s}$: $\mathbb{P}^{\text{nat}}(\boldsymbol{s}) \coloneqq \{\boldsymbol{p}^{(0)}, \boldsymbol{p}^{(1)}, \ldots, \boldsymbol{p}^{(A)}\}$, progressively longer prefixes preceding $\boldsymbol{s}$ in the original training document, where $\boldsymbol{p}^{(0)} = \varnothing$ is the empty string and the longest prefix has length A (Carlini et al., 2023; Cooper et al., 2026a). More generally, a prompt class is a search space: the set of prompts an extraction procedure can navigate to find queries that elicit the candidate. A **prompt-selection rule**

$$f_{\mathbb{P}} : \mathbb{V}^\ell \to 2^{\mathbb{V}^*} \tag{13}$$

maps each candidate $\boldsymbol{s}$ to a finite prompt class $\mathbb{P}(\boldsymbol{s}) \coloneqq f_{\mathbb{P}}(\boldsymbol{s}) \subseteq \mathbb{V}^*$.[5]

Scoring procedure. A **scoring procedure** $\mathsf{score} : \mathbb{V}^\ell \to \mathbb{R}$ measures how easily a candidate $\boldsymbol{s}$ is generated under θ and ϕ from the prompts in class $\mathbb{P}(\boldsymbol{s})$, where higher means more easily generated. For a single prompt $\boldsymbol{p}$, the canonical score is the near-verbatim probability of generating $\boldsymbol{s}$, $q_{\theta,\phi}(\mathbb{B}_\varepsilon(\boldsymbol{s}) \mid \boldsymbol{p})$, or an approximation of that probability (e.g., a beam-search-based lower bound (Cooper et al., 2026b)). (This captures verbatim extraction, when $\varepsilon = 0$, and also greedy decoding, for which the score is binary—1 if $\boldsymbol{s}$ is generated verbatim, otherwise 0.) Over a prompt class for candidate $\boldsymbol{s}$, an **aggregation rule** agg collapses per-prompt scores into the candidate's score:

$$\mathsf{score}(\boldsymbol{s}) = \mathop{\mathsf{agg}}_{\boldsymbol{p} \in f_{\mathbb{P}}(\boldsymbol{s})} \mathsf{score}_{\boldsymbol{p}}(\boldsymbol{s}). \tag{14}$$

For instance, an aggregation rule could return the maximum over the prompt class: the prompt that most easily reproduces an output in $\mathbb{B}_\varepsilon(\boldsymbol{s})$.

Verification procedure. A **verification procedure** takes a score and outputs a decision about whether a candidate $\boldsymbol{s}$ is memorized, i.e., $\mathsf{verify} : \mathbb{R} \to \{\text{memorization claim}, \text{no memorization claim}\}$. We construct such a verify: a conformal test that calibrates a threshold on the score by comparing to the scores of matched non-members, determining when a score is high enough to claim memorization (§3). Put differently, the score this test evaluates is a generation probability, and clearing the calibrated threshold is what licenses calling it an extraction probability.

Putting all five components together:

Definition B.2 (Extraction procedure)**.** An **extraction procedure** is a tuple θ, ϕ, $f_{\mathbb{P}}$, score, verify, where model θ and decoding policy ϕ configure the distribution $q_{\theta,\phi}$; $f_{\mathbb{P}}$ specifies the prompts used for each candidate sequence $\boldsymbol{s}$ (Eq. 13); score returns the score for $\boldsymbol{s}$ (computed under the fixed model θ, decoding policy ϕ, and prompt-selection rule $f_{\mathbb{P}}$); and, based on its score, verify either deems $\boldsymbol{s}$ memorized or makes no claim.

We later show that prior methods in the literature are instantiations of this definition (§H.1).

[5] In practice, each prompt is bounded in length by the model's context window, possibly smaller for specific class constructions (e.g., natural-prefix classes are bounded by the longest prefix of $\boldsymbol{s}$ in the training document).

C EXTENDED DETAILS ON THE CONFORMAL TEST FOR MEMORIZATION

In this work, we use the (near-)verbatim generation probabilities of matched non-training data to calibrate a threshold—a probability cutoff above which (near-)verbatim generation is claimed to be evidence of memorization. Calibration makes the inferential chain behind this claim explicit, which prior work has left implicit (Box 1).

Here, we provide additional details on the conformal test for memorization, which formalizes threshold calibration for making claims about memorization concerning populations, certified to a chosen false positive level. The single-document census procedure answers a different question. It concerns fixed documents, where the question is not what proportion of a sampled population is memorized, but how much of *this* particular work is memorized. As a result, while the census procedure follows the same spirit of matched comparisons for threshold calibration as the conformal test, it's a fundamentally different object. There's no population to sample from in that context, so there is no analogous FPR guarantee to provide. The census threshold $\tau^{(\text{book})}$ is the maximum score over the control book's windows (§E.1). Those windows aren't exchangeable units: using the sliding-window procedure from Cooper et al. (2026a), adjacent windows overlap so they share most of their text; windows from the same paragraph are related even when they don't overlap. So $\tau^{(\text{book})}$ is a measured maximum rather than a rank statement at level $1/(N+1)$ for N the window count.

Concerning the conformal test, here we provide more detail on relevant notation and the setup (§C.1). We provide a full proof of Prop. 1, concerning finite-sample validity of the conformal test (§C.2), and more details on the assumptions and practical limitations of the proved guarantee (§C.3). In particular, that proof requires only exchangeability of the candidate's score and the N control scores. But, in practice, we additionally require that these scores be independent and identically distributed, which our sampling design provides (§4). This lets us provide additional results about the population FPR of the calibrated threshold, from which we're able to provide valid Beta intervals around that FPR, and a guarantee *conditional* on the particular control sample our experiments use, not just a marginal one (§C.4). We also clarify why we can reuse the same control pool across claims for different candidates (§C.5).

C.1 NOTATION AND SUMMARY OF RELEVANT QUANTITIES

There are several different FPR objects related to the conformal test. Here, we walk through exactly what each one is, so that it's very clear what the conformal guarantee is about. Fig. 8 provides a visualization of what this section describes in prose. There are four key objects to distinguish: the target FPR, the certified FPR, the in-sample FPR, and the population FPR. We describe them all, and then how they relate to the calibrated threshold. We'll use our results on Wikipedia as a running example.

The target and certified FPR. As noted in §3, for the conformal test, one picks a **target FPR** α for calibration. From this, the test derives a **conformal** p**-value** denoted $\widehat{p}$ (Eq. 4); for a candidate sequence $\boldsymbol{s}$, we claim memorization when $\widehat{p}(\boldsymbol{s}) \leq \alpha$. Empirically, the p-value comes from a rank cutoff in the size-N control sample. As a result, it may not be possible to obtain exactly the FPR α in practice; it depends on what N is. A finite control sample can only certify an FPR in steps of $1/(N+1)$, because the cutoff has to be one of the N control scores. In other words, the rank $K = \lfloor \alpha(N+1) \rfloor$ provides the cutoff, from which we derive the largest attainable **certified FPR** $\alpha_N = K/(N+1) \leq \alpha$. For instance, consider a control sample from Wikipedia where $N = 5000$; for $\alpha = 0.1\%$, $\alpha_N = 5/5001 \approx 0.09998\%$ (Fig. 8b). This is also why $1/(N+1)$ is the smallest level at which the test can claim anything: since $\widehat{p} \geq 1/(N+1)$, setting $\alpha < 1/(N+1)$ gives $K = 0$, and no memorization claim is ever made (the test can never reject). The floor of $1/(N+1)$ is on how strong a claim the controls can support, not on the certified FPR; a test that never makes a claim has no false positives (but it also doesn't have any use).

The conformal guarantee in Prop. 1 is ultimately about finding an $\alpha_N \leq \alpha$ (§C.2); it's a guarantee about the procedure, and has to do with a count on the number of controls in the sample. This means that α_N is pinned by construction—the rank at which you specify the cutoff for the threshold.

In-sample FPR. The certified FPR is guaranteed with respect to randomness across control draws (and so the stated guarantee is with respect to random variables for the control sequence, W_i). So it's important to realize that the guaranteed FPR is not the same as the actual **in-sample FPR** that

comes from a *specific* drawn control pool (for which we use $\boldsymbol{w}$ in Fig. 2 to indicate a concrete control sequence). To see where this rate comes from, apply the test's rule to the actual drawn controls. For $K \geq 1$ the threshold is the Kth-largest control score, and a sequence is flagged only if it strictly exceeds that score (Eq. 6). Absent ties at the cutoff, the controls that clear it are exactly those ranked $1, \ldots, K-1$, giving an in-sample FPR of $(K-1)/N$. (There are only N drawn controls, which is why that's the in-sample denominator). At $K = 0$ there is no Kth-largest score, so there's no threshold and nothing gets flagged.

For example, for a control pool of $N = 5000$ concrete sequences, we set a threshold from those controls and a chosen α, and then count how many actual control sequences in the pool sit above the threshold—e.g., for $K = 5$, the in-sample FPR is $4/5000 = 0.08\%$ (compare to $\alpha_N = K/(N{+}1) = 5/5001$, which reserves a slot for the fresh candidate among $N{+}1$ exchangeable scores). This is also why the in-sample FPR is 0% whenever $K = 1$, even though the certified FPR there is $1/(N{+}1)$, not 0. Another way to think of this is that, for a fixed N and chosen α, neither α_N nor the in-sample FPR carries any randomness: both are pinned by the rank K before any control is drawn. Two runs of this experiment on entirely different control samples would report the same in-sample FPR, because τ_{α_N} is defined as the Kth-largest control score. What does change across draws is the population FPR α^*.

Population FPR. All three of the above quantities are different from the **population FPR** α^*. To continue our Wikipedia example, this is the FPR over all Wikipedia entries. This is the quantity we actually wish we could know; it's $\alpha^* \coloneqq \mathrm{FPR}(\tau_{\alpha_N})$, the value at our calibrated threshold of the function $\mathrm{FPR}(\cdot)$ that maps each threshold to the fraction of the non-member population scoring above it. That function is fixed and unknown, determined by the model, the score, and the population, with no sampling involved; it exists whether or not anyone ever runs an experiment. α^* is one point on that curve. The curve doesn't change, but the threshold at which we read it changes across control sample draws, and so does α^*.

To make this concrete: you draw N controls, and they get scores. You sort those scores and take the Kth largest one, which gives you τ_{α_N}. There's no other input here, e.g., if $\alpha = 0.1\%$ and $N = 5000$, we have $\alpha_N = 0.09998\%$ and $K = 5$, so τ_{α_N} is the 5th-largest of the 5000 control scores. For instance, for Wikipedia and 50-token suffixes, $\tau_{\alpha_N} = 8.93 \times 10^{-6}$; α^* is the FPR curve's height at $8.93{\times}10^{-6}$ (Fig. 8a). Now say you draw a different set of $N = 5000$ controls from the same population: their 5th-largest probably won't be exactly $8.93{\times}10^{-6}$; maybe it's $7.5{\times}10^{-6}$. The curve hasn't changed, but you end up reading that curve at a different horizontal position τ_{α_N}, and so you get a different height α^*. And the reason the position moves at all on the curve is because the 5th-largest score of a random sample is itself random. So once the controls are drawn, α^* is a fixed but unknown number; across control draws it's random, and it inherits that randomness entirely through τ_{α_N}.

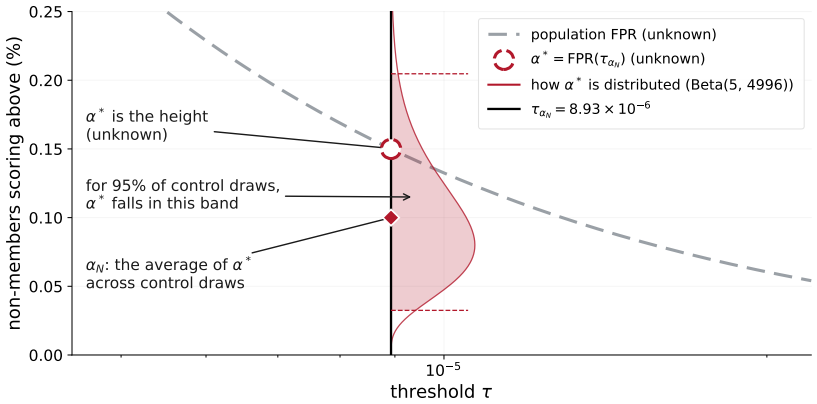
This isn't something the conformal test gives us. Prop. 1 assumes only exchangeability, and it bounds the probability of falsely claiming a non-member is memorized; it says nothing about the value of α^*. If we additionally assume the scores are independent and identically distributed—which our sampling design provides (§4)—as well as continuous, then the probability of that false claim is the average of α^* over control samples (§C.4). (Continuity is the assumption that doesn't come for free: under a top-k policy many continuations may have probability exactly zero, so control scores can tie, and Lemma 3 recovers the result in that case.) Under those assumptions we know its mean, which is fixed as soon as we choose the target α and the number of controls N:

$$\mathbb{E}[\alpha^*] = \alpha_N \leq \alpha. \tag{15}$$

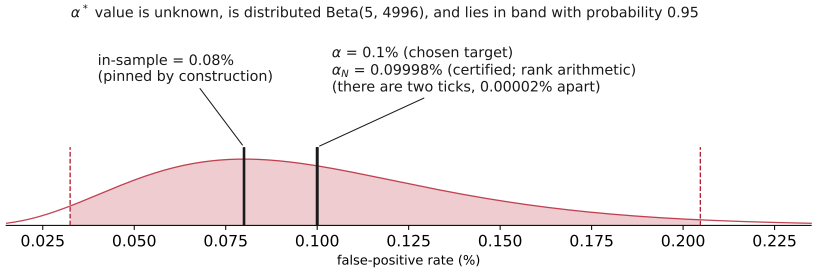
This is what makes α_N the right quantity to subtract from the member generation rate $\hat{G}$ to compute the empirical estimate of calibrated memorization (Eq. 8, $\hat{M} \coloneqq \hat{G} - \alpha_N$). We subtract off the expected value of the population rate (which we don't see directly), so $\hat{G} - \alpha_N$ is unbiased for the procedure-level calibrated rate.

We also know how it's distributed (Vovk, 2012): the Beta law (§C.4.1). The Beta interval on the population FPR (e.g., $[0.032\%, 0.205\%]$) is fixed and known in advance. It depends only on control sample size N and rank K. This means the interval is distribution-free, and so it's the same for both the 10-token and 50-token suffix experiments using the same N and K, even though their thresholds differ by five orders of magnitude (0.95 versus 8.93×10^{-6}). But α^* varies, and falls inside it (e.g., for 95% of control draws for the prior interval example). The specific *quantity* is therefore conditional on the control draw we got, while its *law* is a statement over control draws. This is *not* a confidence interval because the interval itself is *not* computed from data and does *not* vary

across draws—the reverse of the usual situation, where the interval moves and the parameter stays the same. (In contrast, Clopper-Pearson intervals on the generation rate $\hat{G}$ are confidence intervals, computed from a member or non-member sample.)



(a) Population FPR curve (illustrative) and the calibrated τ_{α_N}



(b) The four FPR quantities on one axis

Figure 8: **Relating the four FPR quantities.** We use results from §4 to illustrate: Wikipedia and OLMo 2 32B base, 50-token natural prefixes as prompts and their corresponding 50-token suffixes as the targets for extraction; $N = 5000$ matched non-member controls and target FPR $\alpha = 0.1\%$. This allows for at most $K = 5$ false positives, so the certified level is $\alpha_N = K/(N+1) = 5/5001$ and the threshold τ_{α_N} is the 5th-largest control score (Prop. 1; proof in §C.2). (**a**) The population FPR curve $\mathrm{FPR}(\tau)$ is fixed by the extraction procedure (Def. B.2) and the non-member distribution, but computing it needs that whole distribution; in practice, we have only N draws from it, so it's unknown (dashed; shape illustrative only). Our controls set τ_{α_N}, and $\alpha^* = \mathrm{FPR}(\tau_{\alpha_N})$ is the curve's height there (one point on that unknown curve). We do know α^*'s distribution across control draws: $\alpha^* \sim \mathrm{Beta}(K, N-K+1)$, which here is $\mathrm{Beta}(5, 4996)$ (Vovk, 2012) (§C.4.1). The red curve is that distribution (drawn along the threshold line); its shaded part is where α^* falls for 95% of draws (since α^* is unknown, the dashed circle is drawn at an arbitrary point inside the 95% band). The diamond marks the mean, which is exactly the certified level α_N (Eq. 15). (**b**) All four FPR quantities on one axis, for this same τ_{α_N}: the target α and the in-sample rate, which (a) does not show, alongside α_N and the distribution of α^* (the same as in (a), laid along the axis). The 95% band for α^* runs from 0.03% to 0.20%; the other three values sit within 0.02 percentage points of one another. α^* is the one that depends on the control draw (it varies across draws and is unknown for our specific draw), so it can't be marked. The other three and the band are fixed once α and N are chosen, before any control is drawn. This is why the band isn't a confidence interval: the interval is fixed, α^* isn't.

object	example value	where the value comes from
target α	0.1%	we choose it freely
certified α_N	$5/5001 = 0.09998\%$	a consequence of N and α; what Prop. 1 actually certifies; e.g., $K = \lfloor 0.001 \times 5001 \rfloor = 5$, then $K/(N+1)$
in-sample FPR	$4/5000 = 0.08\%$	for $K \geq 1$, the controls above τ are $K-1$ of them, so the rate is $(K-1)/N$; no guarantee attached
population α^*	unknown; $\mathbb{E}[\alpha^*] = \alpha_N$	guarantee is about this, mean is α_N (e.g., 0.09998%); Beta-distributed (law $\mathrm{Beta}(5, 4996)$), e.g., 95% central interval $[0.032\%, 0.205\%]$
threshold $\tau_\alpha = \tau_{\alpha_N}$	0.95 (10-token) / 8.9×10^{-6} (50-token)	the K-th-largest control score e.g., $K = 5$

Understanding the calibrated threshold. Because the threshold is set via a rank, it's important to realize that $\tau_\alpha = \tau_{\alpha_N}$ (which is why we just use τ_{α_N} in the main text). Consider an arbitrary $x \in (0, 1)$. The subscript notation τ_x means "run the recipe with x as the input": first $K = \lfloor x(N+1) \rfloor$, then take the Kth-largest control score. Supplying the target $\alpha = 0.1\%$ as x with $N = 5000$ gives $K = \lfloor 5.001 \rfloor = 5$, so $\tau_\alpha = \mathsf{score}_{(5)}$, and $\alpha_N = K/(N+1) = 5/5001 \approx 0.09998\%$. Now instead, supply the attained certified level α_N as x. This gives $K = \lfloor (5/5001)(5001) \rfloor = \lfloor 5 \rfloor = 5$, so $\tau_{\alpha_N} = \mathsf{score}_{(5)}$, as well. In other words, both yield the same order statistic (literally the same number). This is always the case, because $\lfloor . \rfloor$ is the only step that discards information, and α_N's computation already drops that information. Applying the floor again does nothing; mapping a target FPR to its attained certified level is idempotent. The same idempotence applies to the test itself: $\widehat{p}$ takes values on the $1/(N+1)$ grid, so $\widehat{p}(\boldsymbol{s}) \leq \alpha$ exactly when $\widehat{p}(\boldsymbol{s}) \leq \alpha_N$—testing at the target FPR and testing at the certified FPR are the same test.

The two thresholds are the same number, but the two subscripts don't record the same information. The map $\alpha \mapsto \tau_\alpha$ is *many-to-one*: every $\alpha \in [5/5001,\ 6/5001) = [0.09998\%,\ 0.11997\%)$ produces the identical threshold $\mathsf{score}_{(5)}$, so τ_α names a threshold by an input that could have been any of the values in that continuum. But the map $\alpha_N \mapsto \tau_{\alpha_N}$ is *one-to-one* on the $1/(N+1)$ grid: each attained level corresponds to exactly one order statistic. So, in summary, τ_α records what was asked for, τ_{α_N} records what was obtained, and α_N is the smallest target in the interval that yields this threshold.

C.2 Finite-sample validity of the conformal test

We prove Prop. 1, including the treatment of ties and the finite-control resolution. The proof follows directly from the canonical results on finite-sample validity of the conformal test from Shafer & Vovk (2008).

Proposition 1 (Finite-sample validity of the conformal test)**.** *Fix $\alpha \in (0, 1)$ and the complete extraction procedure independently of the calibration controls and candidate scores. Let S be a fresh candidate that satisfies the null—a non-member ($\mathsf{membership}_\varepsilon(S) = 0$) drawn by the same rule as the controls—and suppose its score and the N control scores are exchangeable. Then*

$$\Pr_0[\widehat{p}(S) \leq \alpha] \leq \alpha_N \leq \alpha,$$

where $\Pr_0$ is probability under the null, taken over both the random controls and the fresh candidate, so the guarantee is marginal and the left-hand side is the FPR. For $K \geq 1$ (Eq. 5), rejection is exactly the strict threshold test in Eq. 6; for $K = 0$, the deterministic test can't reject. If the $N+1$ scores have no ties, the first inequality is an equality; ties can only make this test conservative.

Setup. Fix the target FPR α and the complete extraction procedure (Def. B.2) before any control or candidate score is observed. In particular, this fixes the model θ, decoding policy ϕ, prompt-selection rule $f_{\mathbb{P}}$ (Eq. 13), scoring function score (Eq. 14), and the form of the conformal verify. The one thing not fixed in advance is the threshold τ_{α_N}, which verify calibrates from the control scores.

Let

$$Z_i = \mathsf{score}(W_i) \quad (i = 1, \ldots, N), \qquad Z_{N+1} = \mathsf{score}(S),$$

where S is a fresh non-member candidate drawn from the matched-non-member null. The assumption is that $(Z_1, \ldots, Z_{N+1})$ is exchangeable: their joint distribution is unchanged by any permutation of the $N+1$ indices. Exchangeability doesn't require a parametric model for their common distribution, or even independence. Our experiments do draw controls i.i.d., which implies exchangeability; the Clopper–Pearson intervals in §4 and the Beta law in §C.4.1 additionally require independence.

Proof. Let $U_1, \ldots, U_{N+1}$ be i.i.d. continuous random variables, independent of the scores $Z_1, \ldots, Z_{N+1}$. Order the pairs (Z_i, U_i) in decreasing order: larger scores come first, and larger U_i breaks a score tie.

The pairs remain exchangeable and are almost surely distinct. Both halves are immediate. The U_i are i.i.d. and independent of the scores, so permuting the $N+1$ indices permutes both coordinates together and leaves the joint distribution unchanged. And the U_i are also continuous, so $\Pr[U_i = U_j] = 0$ for $i \neq j$ and no two pairs tie; the decreasing order is therefore a strict total order, even when scores tie. Because the pairs are almost surely distinct, each of the $N+1$ positions in this order holds exactly one pair, so the candidate's descending rank R is well defined. And because the pairs are exchangeable, swapping index $N+1$ with any index i leaves the joint distribution unchanged, so the candidate is equally likely to occupy any position. The positions are exhaustive and mutually exclusive, so R is uniform:

$$\Pr[R = r] = 1/(N+1) \tag{16}$$

for every $r \in \{1, \ldots, N+1\}$.

We write

$$R = 1 + \sum_{i=1}^{N} \mathbf{1}\{Z_i > Z_{N+1}\} + \sum_{i=1}^{N} \mathbf{1}\{Z_i = Z_{N+1},\, U_i > U_{N+1}\}.$$

That is, R counts the candidate itself, plus the controls that strictly beat its score, plus the controls that tie its score but win the tie-break.

Define the randomly tie-broken rank value among the $N+1$ scores as

$$p^\star = R/(N+1).$$

The deterministic conformal p-value (Eq. 4) counts *all* control ties against the candidate:

$$\widehat{p}(S) = \frac{1 + \sum_{i=1}^{N} \mathbf{1}\{Z_i \geq Z_{N+1}\}}{N+1}.$$

So R and the numerator of $\widehat{p}$ agree on everything except the ties: on the tied controls, R counts only the tie-break winners; $\widehat{p}$ counts all of them. The tie-break winners are a subset of the tied controls, so the numerator of $\widehat{p}$ is at least R, i.e.,

$$\widehat{p}(S) \geq p^\star. \tag{17}$$

Now let $K = \lfloor \alpha(N+1) \rfloor$ and recall $\alpha_N = K/(N+1)$ (Eq. 5). $\widehat{p}(S) \leq \alpha$ holds exactly when $(N+1)\widehat{p}(S) \leq \alpha(N+1)$. The left-hand side is an integer $(1 + \sum_{i=1}^{N} \mathbf{1}\{Z_i \geq Z_{N+1}\})$, so this holds exactly when $(N+1)\widehat{p}(S) \leq \lfloor \alpha(N+1) \rfloor = K$. And $(N+1)\widehat{p}(S) \geq R$ by Eq. 17, so $(N+1)\widehat{p}(S) \leq K$ means $R \leq K$.

Together,

$$\{\widehat{p}(S) \leq \alpha\} = \{(N+1)\widehat{p}(S) \leq K\} \subseteq \{R \leq K\}.$$

The inclusion and uniformity of R (Eq. 16) therefore give

$$\Pr_0[\widehat{p}(S) \leq \alpha] \leq \Pr[R \leq K] = \frac{K}{N+1} = \alpha_N \leq \alpha.$$

If the scores are almost surely distinct, the tie sum vanishes and $\widehat{p} = p^\star$. The event inclusion is then equality, so the false-positive probability is exactly α_N. When ties occur, counting all of them against the candidate makes $\widehat{p}$ weakly larger than its tie-broken counterpart (Eq. 17) and can only remove rejections. Therefore, the deterministic test is conservative.

Rejection has so far been expressed as a condition on the candidate's rank among the $N+1$ scores. It remains to express it as a condition on a cutoff: a comparison of the candidate's score to the Kth-largest control score (Eq. 6).

For $K \geq 1$, sort the control scores in decreasing order, $Z_{(1)} \geq \cdots \geq Z_{(N)}$, and write $C = \left|\{i : Z_i \geq Z_{N+1}\}\right|$ for the number of controls scoring at least as high as the candidate. Recall from above that $\widehat{p}(S) \leq \alpha$ holds exactly when

$$(N+1)\widehat{p}(S) = 1 + C \leq \lfloor \alpha(N+1) \rfloor = K.$$

So,

$$\begin{aligned} \widehat{p}(S) \leq \alpha &\iff 1 + C \leq K \\ &\iff C \leq K - 1 \\ &\iff Z_{N+1} > Z_{(K)} = \tau_{\alpha_N}. \end{aligned}$$

The last step follows from counting positions in the sorted order:

- One direction assumes $Z_{N+1} > Z_{(K)}$ and concludes $C \leq K - 1$: this follows because every control counted in C is then strictly above $Z_{(K)}$, so it sits in one of positions $1, \ldots, K-1$.
- The other direction assumes $C \leq K - 1$ and concludes $Z_{N+1} > Z_{(K)}$. This follows by the contrapositive: if $Z_{N+1} \leq Z_{(K)}$, then $Z_{(1)}, \ldots, Z_{(K)}$ are all at least as large as the candidate score Z_{N+1}, so $C \geq K$ and $C \leq K - 1$ fails. The hypothesis $Z_{N+1} \leq Z_{(K)}$ includes $Z_{N+1} = Z_{(K)}$, so the equality case is one where the test doesn't reject. This is why the last line is strict: a candidate whose score ties the threshold fails.

Finally, if $K = 0$, the numerator of $\widehat{p}$ is at least one, so rejection is impossible. Equivalently, the smallest attainable conformal value is $1/(N+1)$. The effective level α_N therefore changes in increments of $1/(N+1)$: the tightest nonzero level occurs at $K = 1$, when the candidate must strictly exceed the largest control score. For example, for $N = 5{,}000$ and $\alpha = 0.1\%$, $K = \lfloor 5.001 \rfloor = 5$ and $\alpha_N = 5/5001 \approx 0.09998\%$. □

The probability $\Pr_0[\widehat{p}(S) \leq \alpha]$ bounded in Prop. 1 averages over both the N random calibration controls and the fresh non-member candidate. The guarantee is therefore marginal, not conditional on the controls: conditioning on a particular realized control sample generally destroys the rank symmetry used in the proof. The population FPR α^* varies across control sample draws; its distribution is given in §C.4.1.

Likewise, applying the test to many candidates gives this marginal guarantee to each candidate, but doesn't by itself bound the probability of one or more false claims or the false-discovery rate (§C.5, Bates et al. (2021)). This marginal, per-candidate guarantee is what the test is designed to provide, so neither is a shortcoming of the construction.

C.3 Practical limits of matching

The conformal guarantee rests on two conditions (§3): the controls must be exchangeable with the candidate under the matched-non-member null, and they must actually be non-members. Exchangeability requires the controls to be selected by the same rule that would select the candidate if it were a non-member and to have the same source, format, length, and other score-relevant features. However carefully controls are matched, some distribution shift can remain, so in practice one must make a best effort at exchangeability. The calibrated claim is always relative to the resulting null: the certified FPR α_N is a statement about the control distribution, so if a candidate comes from too different a distribution, it isn't the FPR for candidates like it.

Genuine non-members are also difficult to identify because modern training corpora are enormous and typically are of uncertain composition (Lee et al., 2023b). Post-cutoff sources are useful, but they don't solve the problem automatically: Wikipedia entries can reuse pre-cutoff text, and a post-cutoff book can quote public-domain or earlier material that was almost certainly in training. Those sequences are actually members, rather than valid controls.

Both experiments therefore inspect the high-scoring control tail and verify reuse. For illustrative purposes, in the main text, we do *not* remove identified members from the controls in the population experiments (§4); this is so that we can make clear that such contamination of the null raises the

threshold, and so depresses memorization claims. We detail this further in §D, where we do decontaminate for the same (and additional) experimental settings, which shows how removing the verified contaminants changes the calibrated rate reported in the main text. In the census experiments, we do decontaminate: we remove verified contaminants before setting each book's threshold (§5).

To sum up, distribution shift and null contamination cause issues in different directions. Distribution shift between controls and candidates can mean a looser null than intended: if the controls score lower than matched non-members would, the threshold calibrated from them is too low, more non-members clear it, and the actual FPR (if we'd had a better-matched null) could exceed the certified level α_N. In contrast, contamination only makes the test more conservative: a contaminated control that scores high raises the threshold (§D). Calibrated claims should therefore be qualified accordingly (§7); there should be clear reporting of how the controls were selected and matched, and their decontamination status, rather than treating a post-cutoff date as sufficient evidence of non-membership. This matters for recent work claiming that extraction is unreliable evidence of memorization because models can verbatim complete text they weren't explicitly trained on (Liu et al., 2025; Morris et al., 2025). Neither Liu et al. (2025) nor Morris et al. (2025) removes near-duplicates of training data from the sequences they treat as non-members. Without proper decontamination, the sequences treated as non-members can include near-duplicates of training data, which are members (Eq. 2, Lee et al. (2022)), so a verbatim completion of such a sequence actually reflects memorization, rather than being a counterexample to it (§7).

C.4 The population false-positive rate of the threshold

Throughout this appendix we assume more than Prop. 1 does. Its guarantee needs only exchangeability of the candidate's score and the N control scores (§C.2); the results here also require that those scores be independent and identically distributed, and—for Prop. 2—continuous (a requirement Lemma 3 relaxes). Recall from §C.1 that the certified FPR α_N is fixed by the target FPR α and N alone. However, for different drawn N-sized control samples, that α_N is associated with different thresholds τ_{α_N}. This is because, while α_N is set by *rank* within the controls, τ_{α_N} is set by the *specific scores* in the control sample (and of course, the same rank in different samples can be associated with different values).

Every threshold value is associated with a true population FPR: the gray curve in Fig. 8a. We write α^* for the true FPR for a specific threshold—the value of that curve at τ_{α_N}. Its randomness comes entirely from the control sample: the curve doesn't move, but a different sample gives a different τ_{α_N}, and so a different point on it. So we don't know α^*, even though we do know α_N. We do know its mean. Under the assumptions above, Prop. 1's guarantee—that a fresh non-member is falsely claimed memorized with probability at most α_N, over both the control draw and the candidate—becomes a statement about α^*: α_N is the average of α^* over control samples, i.e., $\mathbb{E}[\alpha^*] = \alpha_N$ (Eq. 15). But that constrains only the mean, and says nothing about how far any one draw's α^* sits from it. The rest of this appendix supplies the spread: α^* follows a Beta distribution, and the 95% band in Fig. 8 is where that distribution puts α^* for 95% of samples.

Here, we show this precisely. We translate results from Vovk (2012) into our notation, and make every step explicit. That is, we state and prove the Beta distribution of α^* (Prop. 2), and show how the 95% band in Fig. 8 follows from it. We then extend that result to the score used in an extraction procedure (Eq. 14). The original result assumes that ties between (arbitrary) scores have probability zero, but that's not always the case for actual extraction procedures. For instance, under a top-k decoding policy, many continuations may have probability exactly zero, in which case many controls have a tied score of 0. Ties don't change anything about Prop. 1: its bound holds whether or not scores tie, and ties can only make the test conservative (§C.2). What they change is the law of α^*, and so the interval around it. (Vovk (2012)'s argument already covers tied scores, but yields only a one-sided bound on that law, with equality when the score is continuous.) Lemma 3 shows that, as long as the threshold falls among the nonzero scores, the Beta law still holds exactly on a range of α^* values that, at the level we illustrate ($K = 5$, $N = 5000$), contains the whole 95% band—so the intervals we report are exact. We can also read the same law as a guarantee conditional on the drawn control sample (§C.4.3), which the marginal guarantee alone doesn't provide.

C.4.1 THE BETA LAW

Setup. Notation follows the proof of Prop. 1 in §C.2: $W_1, \ldots, W_N$ are the N matched non-member controls and S is the fresh non-member candidate; $Z_i = \mathsf{score}(W_i)$ is the ith control's score, and $Z_{(K)}$ is the Kth-largest of those N scores, which is the calibrated threshold τ_{α_N}. But there's one deliberate exception. That proof writes Z_{N+1} for the fresh non-member candidate's score, because its argument is about the candidate's rank among the $N+1$ exchangeable scores; the index on the score is useful for calling attention to this. Here, we don't need a rank argument, so we instead write the fresh non-member candidate's score as Y. This keeps it visually distinct from the controls, but Z_{N+1} and Y are the same object.

Proposition 2 (Distribution of the calibrated threshold's population FPR)**.** *Fix $\alpha \in (0,1)$ and the complete extraction procedure independently of the calibration controls and candidate scores. Let $Z_i = \mathsf{score}(W_i)$, $i = 1, \ldots, N$, be the control scores and let $Y = \mathsf{score}(S)$ be the score of a fresh non-member candidate. Suppose the $N+1$ scores $Z_1, \ldots, Z_N, Y$ are mutually independent and identically distributed, sharing the cumulative distribution function (CDF) $F_0(z) = \Pr[Y \le z]$—the fraction of the non-member population whose score is at or below z—and suppose F_0 is continuous.*

Let $K = \lfloor \alpha(N+1) \rfloor \ge 1$ and let $\tau_{\alpha_N} = Z_{(K)}$ be the Kth-largest control score. Then the population FPR of the calibrated threshold—the fraction of the whole non-member population scoring strictly above τ_{α_N} (Eq. 6)—is

$$\alpha^* \coloneqq \mathrm{FPR}(\tau_{\alpha_N}) = 1 - F_0(\tau_{\alpha_N}),$$

and it satisfies

$$\alpha^* \sim \mathrm{Beta}(K,\, N - K + 1),$$

so in particular

$$\mathbb{E}[\alpha^*] = \frac{K}{N+1} = \alpha_N.$$

Proof.
Step 1: α^* as a function of the controls.
By definition of $\mathrm{FPR}(\cdot)$ at a fixed threshold τ,

$$\mathrm{FPR}(\tau) = \Pr[Y > \tau] = 1 - \Pr[Y \le \tau] = 1 - F_0(\tau). \tag{18}$$

This holds for a threshold fixed in advance, but $\tau_{\alpha_N} = Z_{(K)}$ is random; it depends on which controls we drew. Conditioning on the controls resolves this, for two reasons. First, $\tau_{\alpha_N} = Z_{(K)}$ is a function of controls $(Z_1, \ldots, Z_N)$ alone, so once we condition on the controls it's a fixed number. Second, Y is independent of $(Z_1, \ldots, Z_N)$, so conditioning on them doesn't change its distribution: for every z,

$$\Pr[Y \le z \mid Z_1, \ldots, Z_N] = \Pr[Y \le z] = F_0(z).$$

Conditional on the controls, we may therefore apply the identity in Eq. 18 at the now-fixed threshold $\tau_{\alpha_N} = Z_{(K)}$:

$$\Pr[Y > \tau_{\alpha_N} \mid Z_1, \ldots, Z_N] = 1 - F_0(\tau_{\alpha_N}) = 1 - F_0(Z_{(K)}) = \alpha^*$$

which is precisely how α^* was defined (Prop. 2). So $\alpha^* = 1 - F_0(Z_{(K)})$ is a function of the controls alone. But we still need to find its distribution when the controls are redrawn.

Step 2: α^* is the Kth-smallest of N independent uniforms.
Set $V_i = 1 - F_0(Z_i)$ for $i = 1, \ldots, N$. Because F_0 is continuous, $F_0(Z_i)$ is uniform on $[0,1]$ by the probability integral transform, and therefore so is $V_i = 1 - F_0(Z_i)$. The V_i are functions of the independent $Z_1, \ldots, Z_N$, so they are independent.

Let $Z_{(1)} \ge \cdots \ge Z_{(N)}$ be the controls in decreasing order, as in §C.2. Because $1 - F_0$ is nonincreasing, $z \ge z'$ implies $1 - F_0(z) \le 1 - F_0(z')$. So

$$1 - F_0(Z_{(1)}) \le 1 - F_0(Z_{(2)}) \le \cdots \le 1 - F_0(Z_{(N)})$$

is an arrangement of $V_1, \ldots, V_N$ in nondecreasing order.

Write $V_{[1]} \le \cdots \le V_{[N]}$ for the V_i in increasing order. Applying $1 - F_0$ reverses the control score ordering: the largest control score $Z_{(1)}$ gives the smallest value $1 - F_0(Z_{(1)})$, the second-largest

gives the second-smallest, and so on. So the Kth-largest control score $Z_{(K)}$ gives the Kth-smallest V, $V_{[K]}$. That is,

$$\alpha^* = 1 - F_0(Z_{(K)}) = V_{[K]}.$$

Step 3: the distribution function of α^*.
Fix $u \in [0, 1]$ and let $C_u = \sum_{i=1}^{N} \mathbf{1}\{V_i \le u\}$ count the V_i that are at most u.

First, because the $V_{[1]} \le \cdots \le V_{[N]}$ are sorted, the V_i that are at most u are exactly the first C_u of them: $V_{[1]}, \ldots, V_{[C_u]}$. So $V_{[K]}$ is among them (i.e., $V_{[K]} \le u$) exactly when $C_u \ge K$.

Second, $C_u \sim \text{Binomial}(N, u)$. This is because each V_i is uniform on $[0, 1]$, so $\Pr[V_i \le u] = u$. And the V_i are independent (Step 2), so the indicators $\mathbf{1}\{V_i \le u\}$ are independent as well. A sum of N independent indicators, each equal to 1 with probability u, is by definition $\text{Binomial}(N, u)$.

Putting these together with $\alpha^* = V_{[K]}$ (Step 2),

$$H(u) \coloneqq \Pr[\alpha^* \le u] = \Pr[C_u \ge K] = \sum_{j=K}^{N} \binom{N}{j} u^j (1-u)^{N-j},$$

the binomial probability of at least K successes.

Step 4: H is the distribution function of $\text{Beta}(K, N - K + 1)$.
For every $u \in [0, 1]$, the binomial tail $\sum_{j=K}^{N} \binom{N}{j} u^j (1-u)^{N-j}$ equals the $\text{Beta}(K,\, N - K + 1)$ distribution function at u.

This is the standard binomial–Beta identity from Lemma 3 in Vovk (2012). So H is the Beta distribution function, and therefore

$$\alpha^* \sim \text{Beta}(K,\, N - K + 1).$$

Step 5: the mean is α_N.
A $\text{Beta}(a, b)$ variable has mean $a/(a + b)$. With $a = K$ and $b = N - K + 1$ the two parameters sum to $N + 1$, so

$$\mathbb{E}[\alpha^*] = \frac{K}{K + (N - K + 1)} = \frac{K}{N + 1} = \alpha_N.$$

□

The result in Prop. 2 isn't new: it's a classical fact about rank-based thresholds, stated by Vovk (2012, Proposition 2b and Lemma 3). We derive it in our own notation because the paper's variables map onto ours with a sign change that is easy to get wrong. The one step we take from Vovk (2012) rather than derive is the binomial–Beta identity (Step 4). (In that paper, a conformity score is *high* when an example conforms, so his score is the negative of our extraction score and his kth-*smallest* calibration score is our Kth-*largest* control score; his ε, n, and k are our α, N, and K, with $k = \lfloor \varepsilon(n + 1) \rfloor = K$; his "error"—the true label falling outside the prediction set—is our false positive for memorization; and the symbol α^* in his proof denotes a quantile of the score distribution, not the population FPR we use α^* for here.)

C.4.2 Our score isn't always continuous

Prop. 2 assumes a continuous F_0, but an extraction score doesn't always satisfy this. For instance, under the top-40 decoding policy, a natural suffix whose target tokens are not all in the top 40 has probability exactly zero. (This is the case for the results in Fig. 4a: 89.6% of the 50-token control scores are exactly 0; 41.1% at 10 tokens). A value like this—one that the score takes with strictly positive probability—is an atom of the score distribution. An atom at z is a jump in F_0 at z, of height $\Pr[Y = z]$; a score distribution is continuous precisely when it has no atoms anywhere.

The definition $\alpha^* = 1 - F_0(\tau_{\alpha_N})$ (Prop. 2) doesn't need a continuity assumption; only its Beta distribution does. So the question is, how much of that distribution survives when the score has an atom? The answer depends on the *position* of the atom, not whether an atom exists. The calibrated threshold is one of the highest control scores, while the set of tied zero scores sits at the very bottom

of the range. As long as enough controls score above zero, the threshold lands clear of that block, and the argument behind Prop. 2 goes through.

In terms of the proof of Prop. 2, continuity entered in exactly one place: Step 2, where the probability integral transform makes each $V_i = 1 - F_0(Z_i)$ uniform on $[0, 1]$. The atom is where F_0 fails to be continuous, and all it costs is the uniformity of V_i. $V_i = 1 - F_0(Z_i)$ is just the FPR at control i's score, and FPR falls as that score rises: the higher the threshold, the fewer non-members lie above it (Step 2). So the atom at the bottom of the score range lands at the largest FPR any threshold can produce: every control scoring 0 gets the same $V_i = \alpha^\dagger = \mathrm{FPR}(0)$, the fraction of the non-member population scoring above 0. Below $\alpha^\dagger$—a region only controls scoring above 0 can reach—V_i is still uniform: $\Pr[V_i \le v] = v$ for every $v < \alpha^\dagger$. Fig. 4b annotates $\alpha^\dagger$ as the maximum α_N: 10.4% for 50-token suffixes and 58.9% for 10-token suffixes, in each case the fraction of controls whose score is nonzero. And $\alpha^* = V_{[K]}$ is the Kth-*smallest* of the V_i (Step 2), so whenever at least K controls score above the atom, α^* falls below $\alpha^\dagger$.

The following lemma makes this precise:

Lemma 3 (The Beta law when atoms sit below the calibrated threshold)**.** *We use the same setting of Prop. 2: the $N + 1$ scores $Z_1, \ldots, Z_N, Y$ are mutually independent and identically distributed, sharing the CDF F_0; let $\alpha \in (0, 1)$, $K = \lfloor \alpha(N + 1) \rfloor \ge 1$, $\tau_{\alpha_N} = Z_{(K)}$, and $\alpha^* = \mathrm{FPR}(\tau_{\alpha_N}) = 1 - F_0(Z_{(K)})$, but do* not *assume that F_0 is continuous. Suppose instead that all atoms sit at or below some score $\tau^\dagger$, so $\Pr[Y = z] = 0$ for every $z > \tau^\dagger$. Write*

$$\alpha^\dagger = \mathrm{FPR}(\tau^\dagger) = \Pr[Y > \tau^\dagger] \tag{19}$$

for the share of the non-member population scoring above $\tau^\dagger$, and suppose $0 < \alpha^\dagger < 1$, i.e., some non-members do score above $\tau^\dagger$, and some do not. Then for every $v \in [0, \alpha^\dagger)$,

$$\Pr[\alpha^* \le v] = \sum_{j=K}^{N} \binom{N}{j} v^j (1 - v)^{N-j},$$

which is the distribution function of $\mathrm{Beta}(K,\ N - K + 1)$ evaluated at v—the same distribution Prop. 2 gives under continuity. That accounts for every v below $\alpha^\dagger$—a range α^* can only reach when the threshold is above $\tau^\dagger$; the rest of the probability sits at or above $\alpha^\dagger$, and is exactly $\Pr[\alpha^* \ge \alpha^\dagger] = \Pr[\mathrm{Binomial}(N, \alpha^\dagger) \le K - 1]$, i.e., the probability that fewer than K of the N controls score above $\tau^\dagger$.

Proof.

Step 1: α^* is still the Kth-smallest of the V_i.

Step 2 of the proof of Prop. 2 does two separate things: it identifies α^* as the Kth-smallest of the values $V_i = 1 - F_0(Z_i)$, and it shows that each V_i is uniform. Only the second uses continuity. Here we check that the first still holds. (Step 2 will recover what we need with respect to continuity: each V_i is uniform below $\alpha^\dagger$, even though it's not uniform on all of $[0, 1]$.)

Set $V_i = 1 - F_0(Z_i) = \mathrm{FPR}(Z_i)$, and write $V_{[1]} \le \cdots \le V_{[N]}$ for these values in increasing order. Because $1 - F_0$ is nonincreasing, ordering the controls as $Z_{(1)} \ge \cdots \ge Z_{(N)}$ makes $1 - F_0(Z_{(1)}) \le \cdots \le 1 - F_0(Z_{(N)})$ a nondecreasing permutation of $V_1, \ldots, V_N$. Its Kth entry is therefore $V_{[K]}$, so $\alpha^* = 1 - F_0(Z_{(K)}) = V_{[K]}$.

Nothing above assumes the scores are distinct. Without continuity two controls can score exactly the same value (i.e., there can be an atom), which is why the ordering $Z_{(1)} \ge \cdots \ge Z_{(N)}$ may contain equalities rather than being a strict ranking. Even so, because that ordering is written with $\ge$ and $\le$, the Kth entry of the sorted list is still the Kth smallest.

Step 2: the V_i are uniform below $\alpha^\dagger$.

Continuity fails only where the atoms are, and by hypothesis they all sit at or below $\tau^\dagger$. So the conditional distribution of score Z_i, given that it exceeds $\tau^\dagger$, is continuous; the probability integral transform applies to that distribution, and multiplying by $\alpha^\dagger$ turns the uniform it produces into the statement we want about V_i.

Let $F_0^\dagger$ be the distribution function of Y conditional on $Y > \tau^\dagger$:

$$F_0^\dagger(z) = \Pr[Y \le z \mid Y > \tau^\dagger] = \begin{cases} 0 & z \le \tau^\dagger, \\ \dfrac{F_0(z) - F_0(\tau^\dagger)}{\alpha^\dagger} & z > \tau^\dagger. \end{cases} \tag{20}$$

The numerator is the population mass in $(\tau^\dagger, z]$; dividing by $\alpha^\dagger$, the mass above $\tau^\dagger$, rescales it so that $F_0^\dagger$ reaches 1. Each Z_i shares F_0 with Y, so each shares $F_0^\dagger$ too: $F_0^\dagger$ is also the distribution function of Z_i conditional on $Z_i > \tau^\dagger$.

$F_0^\dagger$ is continuous everywhere. Below $\tau^\dagger$, conditioning on $Y > \tau^\dagger$ gives that whole region probability 0, so $F_0^\dagger$ is flat at 0 there and is continuous. That region is where the atoms are: the conditioning discards exactly the part of the score range where F_0 fails to be continuous. By hypothesis there are no atoms above $\tau^\dagger$, so F_0 is continuous above $\tau^\dagger$. For $z > \tau^\dagger$, $F_0^\dagger(z)$ is $F_0(z)$ shifted and rescaled, so $F_0^\dagger$ is continuous above $\tau^\dagger$ too. At $\tau^\dagger$ itself it is continuous because F_0 is right-continuous.

Therefore, the probability integral transform applies, exactly as it did to F_0 in Step 2 of the proof of Prop. 2. The transform says that if a score has a continuous distribution function F, then F applied to that score is uniform on $[0, 1]$. Here the score is Z_i given $Z_i > \tau^\dagger$, and its distribution function is $F_0^\dagger$. So, conditionally on $Z_i > \tau^\dagger$, $F_0^\dagger(Z_i)$ is uniform on $[0, 1]$.

The uniformity we just proved is attached to $F_0^\dagger(Z_i)$, but we need it for $V_i = \mathrm{FPR}(Z_i)$; we next write V_i as a function of $F_0^\dagger(Z_i)$, so that the uniformity carries over. Fix $z > \tau^\dagger$. Scoring above z already implies scoring above $\tau^\dagger$, so the event $\{Y > z\}$ sits inside $\{Y > \tau^\dagger\}$, and their intersection is just $\{Y > z\}$. Therefore

$$\mathrm{FPR}(z) = \Pr[Y > z] = \Pr[Y > \tau^\dagger] \cdot \Pr[Y > z \mid Y > \tau^\dagger] = \alpha^\dagger\big(1 - F_0^\dagger(z)\big), \tag{21}$$

where the first equality is Eq. 18 at z, and the last substitutes Eq. 19 for the first factor and the complement of Eq. 20 for the second.

Eq. 21 holds for every $z > \tau^\dagger$, so we can put $z = Z_i$ whenever $Z_i > \tau^\dagger$:

$$V_i = \mathrm{FPR}(Z_i) = \alpha^\dagger\big(1 - F_0^\dagger(Z_i)\big).$$

Since $F_0^\dagger(Z_i)$ is uniform on $[0, 1]$ conditionally on $Z_i > \tau^\dagger$, so is $1 - F_0^\dagger(Z_i)$. Multiplying a uniform on $[0, 1]$ by $\alpha^\dagger$ gives a uniform on $[0, \alpha^\dagger]$. So, conditionally on $Z_i > \tau^\dagger$, V_i is uniform on $[0, \alpha^\dagger]$: for every $v < \alpha^\dagger$, $\Pr[V_i \le v \mid Z_i > \tau^\dagger] = v/\alpha^\dagger$.

A control with $Z_i \le \tau^\dagger$ has $V_i = \mathrm{FPR}(Z_i) \ge \mathrm{FPR}(\tau^\dagger) = \alpha^\dagger$, because $\mathrm{FPR} = 1 - F_0$ is nonincreasing (Step 1). So for every $v < \alpha^\dagger$, such a control has $V_i \ge \alpha^\dagger > v$, and therefore never has $V_i \le v$: $\Pr[V_i \le v \mid Z_i \le \tau^\dagger] = 0$. (This conditioning is well defined because $\Pr[Z_i \le \tau^\dagger] = 1 - \alpha^\dagger > 0$.) The two cases above—$Z_i > \tau^\dagger$ and $Z_i \le \tau^\dagger$—are disjoint and exhaustive, so the law of total probability recombines them: for every $v \in [0, \alpha^\dagger)$,

$$\begin{aligned} \Pr[V_i \le v] &= \Pr[Z_i > \tau^\dagger] \cdot \Pr[V_i \le v \mid Z_i > \tau^\dagger] + \Pr[Z_i \le \tau^\dagger] \cdot \Pr[V_i \le v \mid Z_i \le \tau^\dagger] \\ &= \alpha^\dagger \cdot \frac{v}{\alpha^\dagger} + (1 - \alpha^\dagger) \cdot 0 = v. \end{aligned}$$

This is uniformity below $\alpha^\dagger$: the distribution function of V_i is the identity on $[0, \alpha^\dagger)$, which is the range both cases above cover.

Step 3: below $\alpha^\dagger$, the rest of the proof of Prop. 2 applies unchanged.
Steps 3 and 4 of the proof of Prop. 2 need exactly three things from our setting:

- that the V_i are independent, which holds here because they are functions of the independent $Z_1, \ldots, Z_N$
- that $\Pr[V_i \le v] = v$, which Step 2 of this proof supplies for every $v < \alpha^\dagger$
- the identity $\alpha^* = V_{[K]}$, which is Step 1 of this proof

So for every $v \in [0, \alpha^\dagger)$, those Steps 3 and 4 of the proof of Prop. 2 run verbatim, with $C_v = \sum_{i=1}^{N} \mathbf{1}\{V_i \le v\} \sim \text{Binomial}(N, v)$:

$$\Pr[\alpha^* \le v] = \Pr[V_{[K]} \le v] = \Pr[C_v \ge K] = \sum_{j=K}^{N} \binom{N}{j} v^j (1-v)^{N-j} = H(v),$$

the distribution function of $\text{Beta}(K,\ N-K+1)$, where H denotes this polynomial in v.

Step 4: the mass at and above $\alpha^\dagger$.
Step 3 gives $\Pr[\alpha^* \le v] = H(v)$ only for $v < \alpha^\dagger$, so we take the limit of both sides as v increases to $\alpha^\dagger$ from below (written $v \uparrow \alpha^\dagger$).

Take the left side first. Increasing v only adds outcomes, so $\{\alpha^* \le v_1\} \subseteq \{\alpha^* \le v_2\}$ whenever $v_1 < v_2$. Over all $v < \alpha^\dagger$, these events cover exactly $\{\alpha^* < \alpha^\dagger\}$: strictly less than, because no such v reaches $\alpha^\dagger$ itself. The probability of a growing family of events is the limit of their probabilities, so $\Pr[\alpha^* \le v]$ tends to $\Pr[\alpha^* < \alpha^\dagger]$.

Now the right side. H is a polynomial, and so it's continuous. Therefore $H(v)$ tends to $H(\alpha^\dagger)$. Equating the two limits gives $\Pr[\alpha^* < \alpha^\dagger] = H(\alpha^\dagger)$, and therefore

$$\begin{aligned}\Pr[\alpha^* \ge \alpha^\dagger] = 1 - H(\alpha^\dagger) &= \sum_{j=0}^{K-1} \binom{N}{j} (\alpha^\dagger)^j (1-\alpha^\dagger)^{N-j} \\ &= \Pr[\text{Binomial}(N, \alpha^\dagger) \le K-1]. \end{aligned} \tag{22}$$

The sum $\sum_{i=1}^{N} \mathbf{1}\{Z_i > \tau^\dagger\}$ counts the controls scoring above $\tau^\dagger$. Each control does so with probability $\alpha^\dagger$ (Eq. 19), and the Z_i are independent, so that count is $\text{Binomial}(N, \alpha^\dagger)$. Therefore, Eq. 22 says that $\Pr[\alpha^* \ge \alpha^\dagger]$ is the probability that fewer than K of the N controls score above $\tau^\dagger$. □

Remark 4 (An atom exactly at $\tau^\dagger$). The lemma allows atoms "at or below" $\tau^\dagger$. Under the top-40 policy the atom is the block of zero scores, so $\tau^\dagger = 0$ and the atom sits at $\tau^\dagger$ rather than strictly below it. This is exactly the boundary case the lemma's "at or below" allows.

It's therefore worth highlighting that nothing in the proof depends on the atom being strictly below $\tau^\dagger$. Four places could have been sensitive to this case, and each resolves the same way: through a strict inequality.

1. $\alpha^\dagger$ is what $F_0^\dagger$ is normalized by, so it has to be exactly the probability of the event we condition on. The lemma defines $\alpha^\dagger = \Pr[Y > \tau^\dagger]$ with the same strict inequality as that conditioning, so a control scoring exactly $\tau^\dagger$ counts in neither.

2. Step 2 needs $F_0^\dagger$ to be continuous at $\tau^\dagger$, and $F_0^\dagger$ is built from F_0, which is *not*: F_0 rises at $\tau^\dagger$ by the atom's mass $\Pr[Y = \tau^\dagger]$. The strict inequality in the conditioning $Y > \tau^\dagger$ is what keeps that mass out. In the formula it appears as the subtraction of $F_0(\tau^\dagger)$: a distribution function is $\Pr[Y \le z]$, so $F_0(\tau^\dagger)$ already counts the atom's mass, and subtracting it removes it.

3. Step 2 needs controls at or below $\tau^\dagger$ to contribute nothing to $\Pr[V_i \le v]$ when $v < \alpha^\dagger$. A control landing exactly on the atom has $V_i = \text{FPR}(\tau^\dagger) = \alpha^\dagger$, and the lemma's range $[0, \alpha^\dagger)$ stops strictly short of that value, so $V_i \le v$ never holds for such a control. Its contribution is therefore zero: $\Pr[V_i \le v \mid Z_i \le \tau^\dagger] = 0$.

4. Step 4 needs the count $\sum_{i=1}^{N} \mathbf{1}\{Z_i > \tau^\dagger\}$ to be $\text{Binomial}(N, \alpha^\dagger)$, so a single indicator must succeed with probability exactly $\alpha^\dagger$. It does: the Z_i share the distribution of Y, and the indicator tests the same strict inequality that defines $\alpha^\dagger$ (Eq. 19). The strict inequality matters here, since the atom puts positive probability on $Z_i = \tau^\dagger$. Had the indicator tested $Z_i \ge \tau^\dagger$, it would have succeeded with probability $\alpha^\dagger$ plus the atom's mass.

The atom does affect α^* in one way: it gives α^* positive probability of equaling $\alpha^\dagger$ exactly. But the Beta law is claimed only on $[0, \alpha^\dagger)$, and everything from $\alpha^\dagger$ upward is reported as the single number $\Pr[\alpha^* \ge \alpha^\dagger]$, which includes $\Pr[\alpha^* = \alpha^\dagger]$. That is why the lemma splits its two claims at a strict inequality.

C.4.3 CONDITIONAL GUARANTEES

Recall that Prop. 1's guarantee is *marginal*: it averages over control draws. Under our additional assumption that the control scores and the candidate's score are independent and identically distributed (stated at the start of this appendix), this means that the average of the population FPR α^* over control draws is the certified level α_N, i.e., $\mathbb{E}[\alpha^*] = \alpha_N$. Recall that α^* changes with the control draw: it's the population FPR of the calibrated threshold, and that threshold is set from the specific control sample drawn, for the chosen target α. An average of 0.1% is compatible with α^* being 0.05% for half of the control samples and 0.15% for the other half. So, for the specific control sample we draw for our experiments, the marginal guarantee doesn't say how far the specific (unknown) α^* may sit from α_N.

A *conditional* guarantee is a statement about that specific control sample's α^* (Vovk, 2012). Since α^* can't be observed, such a guarantee takes the form of a bound that holds for most control samples: $\alpha^* \le v$ for all but a fraction δ of possible control draws. This statement requires the distribution of α^* across control draws, not just its mean, which is what Prop. 2 and Lemma 3 supply (and what exchangeability alone, as in Prop. 1, can't).

Corollary 5 (A guarantee conditional on the drawn control sample)**.** *Under the assumptions of Prop. 2 for every $v \in (0, 1)$, and under those of Lemma 3 for every $v \in [0, \alpha^\dagger)$,*

$$\Pr[\alpha^* \le v] = \sum_{j=K}^{N} \binom{N}{j} v^j (1-v)^{N-j}. \tag{23}$$

That is, over the control draws, the calibrated threshold set from the drawn control sample has population FPR α^* at most v, with exactly the above probability. Equivalently, for any $\delta \in (0, 1)$, let $v_{1-\delta}$ be the $(1-\delta)$-quantile of $\mathrm{Beta}(K,\ N-K+1)$ (with $v_{1-\delta} < \alpha^\dagger$ under the lemma's assumptions). Then, with probability at least $1-\delta$ over control draws,

$$\alpha^* \le v_{1-\delta}. \tag{24}$$

Proof. Both statements are the distribution function of α^*, evaluated at v and at $v_{1-\delta}$ respectively. Prop. 2 gives that distribution function for every v when F_0 is continuous; Lemma 3 gives it for every $v < \alpha^\dagger$ when F_0 has atoms at or below $\tau^\dagger$. □

To make this concrete, we can walk through the corollary with some of the specific numbers in our Wikipedia experiments. With $N = 5000$ controls and target FPR $\alpha = 0.1\%$, $K = 5$ and the certified FPR is $\alpha_N = 5/5001 \approx 0.1\%$. The certified level α_N is the mean of α^*, but it is not a bound on it: evaluating Eq. 23 at $v = \alpha_N$, α^* exceeds α_N for 44% of control samples. So the statement "the specific calibrated threshold's FPR is at most α_N" holds for only 56% of control samples; it's not a statement one can make with high confidence. Using Eq. 24, what one can say with 90% confidence is that $\alpha^* \le v_{0.9} = 0.16\%$, and with 95% that $\alpha^* \le v_{0.95} = 0.18\%$. (The two-sided 95% interval $[0.032\%, 0.205\%]$ reported in §C.1 comes from the same distribution: its endpoints are $v_{0.025}$ and $v_{0.975}$.)

Next, define the relative gap between the $(1-\delta)$-quantile and the certified level,

$$\rho_{1-\delta} := \frac{v_{1-\delta} - \alpha_N}{\alpha_N}, \tag{25}$$

which depends only on N, K, and δ. For a fixed δ, $\rho_{1-\delta}$ is proportional to $1/\sqrt{K}$.[6] To continue the example, consider also the most conservative operating point, where the calibrated threshold is the top control score ($K = 1$, $\alpha_N = 1/5001 \approx 0.02\%$). There, Eq. 23 gives $\Pr[\alpha^* \le 0.02\%] = 63\%$: the calibrated threshold's population FPR is at most its own certified level for only 63% of control samples. It's at most 0.05% for 92% of samples, and at most 0.1% for 99.3%. A smaller K therefore lowers both α_N and $v_{1-\delta}$, but it raises the relative gap between them: at $K = 1$, $\rho_{0.9} = 1.30$. Because Lemma 3 makes the corollary exact for every $v < \alpha^\dagger$, this reading applies unchanged to the top-40 results of the main text, where $\alpha^\dagger \approx 10.4\%$.

[6] The standard deviation of $\mathrm{Beta}(K,\ N-K+1)$ is approximately $\sqrt{K}/(N+1)$ while its mean is $K/(N+1)$. In the example above ($K = 5$, $\delta = 0.1$), $\rho_{0.9} = 0.60$.

In our figures, we focus on the $K = 5$ operating point. The calibrated rate we report subtracts the certified level, $\hat{M} = \hat{G} - \alpha_N$ (Eq. 8), rather than the unknown α^* (§D.1). We can't subtract α^* itself, since it isn't observable, and subtracting α_N keeps $\hat{M}$ unbiased (§C.1). A conservative alternative would be to subtract a high quantile of its distribution instead, understating the rate by design. Doing so would change little here. Replacing $\alpha_N \approx 0.1\%$ with $v_{0.95} = 0.18\%$ lowers $\hat{M}$ by 0.08 percentage points, roughly a tenth of the width of the interval already plotted around the member curve at that point (Fig. 4b).

It's also worth noting that Prop. 2 assumes every control is a non-member, which our running example shows isn't always the case in practice. In that experiment, a detailed decontamination process revealed that twelve of the $5{,}000$ are actually members (§D.2). Their presence raises the calibrated threshold; removing them raises the calibrated rate from 0.44% to 3.06% at the tightest certified FPR. Since α^* is the fraction of the non-member population above the calibrated threshold, a higher threshold means a smaller α^* than a non-contaminated control pool would have produced. Bounds like $\alpha^* \leq v_{0.95} = 0.18\%$ are therefore conservative here, holding for more than the stated 95% of control draws.

C.4.4 The point of all this in practice

We allow an extraction procedure to choose its decoding policy ϕ, and some policies put an atom in the score distribution. Under top-40 decoding the score is exactly 0 for every suffix the policy cannot generate, and strictly positive otherwise, so $\tau^\dagger = 0$. The atom has one consequence: $\alpha^\dagger = \mathrm{FPR}(0)$, the share of the non-member population the policy can generate at all, is a ceiling on the attainable certified FPR, since no threshold puts more than $\alpha^\dagger$ of that population above it. This is the maximum attainable α_N annotated in Fig. 4b for OLMo 2 32B on Wikipedia: 10.4% for 50-token suffixes and 58.9% for 10-token suffixes.

Importantly, though, this regime (the highest attainable FPR) is actually the *least* interesting for extraction: we care about the *low* FPR regime, which is far away from all the zero-probability controls (the atom). For our relatively small sample size of $N = 5000$, we calibrate at $\alpha_N \approx 0.1\%$. (For larger sample sizes, we'd ideally calibrate to an even lower α_N.) And the lemma says the Beta law is exact for every α^* below $\alpha^\dagger \approx 10.4\%$, which is everywhere our reported intervals live (§C.1). So those intervals are exact, not approximations that ignore the atom.

In practice, we don't have to worry about atoms at nonzero scores. For this to happen, continuations for controls would have to share an identical product of token probabilities, which is extremely unlikely in practice. (Nevertheless, we can check this; in our experiments, the nonzero control scores are all distinct.)

The one case the lemma rules out is $\alpha^\dagger = 0$: the extraction procedure can't generate any non-member at all. There the Beta law has nothing to say, since the interval $[0, \alpha^\dagger)$ it's claimed on is empty. The test itself still works—every control scores 0, so the threshold is 0, and any candidate the procedure can generate is claimed memorized, with a realized false-positive rate of 0. But 0 is now the only attainable FPR: every α_N returns the same threshold and the same rejection region, so the sweep of operating points our figures plot collapses to a point (and so with it the Beta band).

Greedy decoding is the same story, but with two attainable FPRs instead of one: scores take only the values 0 and 1, so every threshold sits on an atom and the only attainable FPRs are $\Pr[Y = 1]$ and 0 (§4). Prop. 1 still holds there—the marginal guarantee never needed continuity—but α^* isn't Beta-distributed and the band shouldn't be drawn. Greedy plots carry error bars (Fig. 3), but those are ordinary Clopper–Pearson confidence intervals on the measured generation rates, computed from the sample—unlike the Beta band on α^*, which is fixed before any control is drawn (§C.1).

C.5 Multiple testing across candidates

The conformal test makes a claim about one candidate at a time: for each candidate it either claims memorization or makes no claim at all. Prop. 1 bounds the probability of claiming memorization for a non-member, taken over both the random controls and the fresh candidate (§C.2). One control sample can therefore serve to make calibrated claims for many candidates; nothing in the guarantee requires those candidates to be independent of one another, so each candidate's claim stands on its own.

Reusing that sample across candidates does, however, make the resulting claims dependent on one another. Every candidate is tested against the same threshold τ_{α_N}, the Kth-largest score in the one control sample we drew (Eq. 6). Its value depends on which controls we happened to draw. So under a different draw, the same candidates would be tested against a different threshold; raising it can only remove claims, and lowering it can only add them. Given a continuous score, take two non-member candidates drawn independently of each other and of the controls, and see whether the test claims memorization for each of them; the two outcomes are positively correlated, at exactly $1/(N+2)$ (Bates et al., 2021). The same $1/(N+2)$ correlation holds for the conformal p-values (Eq. 4) and for any finite-valued function of them, so the dependence can't be transformed away.[7] Each individual claim still carries the guarantee of Prop. 1. But it doesn't bound the error rates that arise when many candidates are tested together—e.g., all 5000 Wikipedia candidates at once.

Marginal error rates across candidates. Suppose the conformal test is applied to candidates $S_1, \ldots, S_J$, where the candidates are random draws of a specified length from data of interest, e.g., all Wikipedia entries (the random counterparts of $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_J$ in Fig. 2). Let $\mathbb{S}_0 = \{j : \mathsf{membership}_\varepsilon(S_j) = 0\}$ index the non-members among them (Eq. 2). As in Bates et al. (2021), we treat $\mathbb{S}_0$ as fixed rather than random: what varies across draws is the candidates' scores, not which of them are non-members. Each S_j with $j \in \mathbb{S}_0$ is then a fresh non-member drawn from the matched control population of interest (exactly the S of Prop. 1). Let $\mathbb{M} = \{j : \mathsf{score}(S_j) > \tau_{\alpha_N}\}$ index the candidates claimed to be memorized (Eq. 6). A false positive is a non-member claimed to be memorized, so $|\mathbb{M} \cap \mathbb{S}_0|$ counts the false positives among the J candidates.

Generally speaking, there are two error rates for false positives across the J candidates that are of interest: the **familywise error rate (FWER)** and the **false discovery rate (FDR)** (Benjamini & Hochberg, 1995). The familywise error rate is the probability of at least one false positive among the J candidates. The false discovery rate is the expected share of false positives among all the candidates claimed to be memorized. More formally,

$$\mathrm{FWER} \coloneqq \Pr\big[\,|\mathbb{M} \cap \mathbb{S}_0| \geq 1\,\big], \qquad \mathrm{FDR} \coloneqq \mathbb{E}\left[\frac{|\mathbb{M} \cap \mathbb{S}_0|}{\max\{1, |\mathbb{M}|\}}\right]. \tag{26}$$

Both of these rates, like the FPR in Prop. 1, are marginal: they average over the draw of the N controls and the candidates. Therefore, they are statements about the conformal test, rather than the specific control sample and set of candidates that an experiment happens to use.

A difference from the FPR is that Prop. 1 concerns a claim for a single candidate (whether or not *that* candidate is memorized), whereas these two rates concern all J candidates' claims together. All three rates depend on the threshold τ_{α_N}, which the controls set (Eq. 6). But the FPR is a probability about a single hypothetical candidate: a fresh non-member, drawn the same way the controls are. It doesn't depend on which candidates we test, or on how many. FWER and FDR, by contrast, are defined over the J candidates we actually test, through $\mathbb{S}_0$ and $\mathbb{M}$ (Eq. 26). So if none of those candidates is a non-member, $|\mathbb{M} \cap \mathbb{S}_0| = 0$ regardless of $\mathbb{M}$, and both rates are 0 by construction. That is the case in our experiments, where every candidate is a member (§4, §D).[8]

[7] Our score isn't always continuous. For instance, under top-k decoding many sequences share a score of exactly 0 (§C.4.2). Bates et al. (2021) prove the exact value $1/(N+2)$ for a continuous score and extend the positive dependence to scores with atoms for a randomized p-value (their Appendix A.3; see the footnote in the BH paragraph below), but don't extend this exact correlation to that setting. We don't pursue this here, either.

[8] This is a different point from the one made in the main text, that the certified FPR (based on the control generation rate) gives the share of the member generation rate that may reflect false positives (§4). In that setting, all candidates are members, so those specific candidates can't literally be false positives with respect to membership. But this is beside the point if we step through the inferential claim (Box 1). We use score as the signal: a generation probability that clears the calibrated threshold is too high to be explained by general predictability, which is what supports the conclusion that the candidate is a member that the model memorized. In this sense, the conformal test is a test for whether a candidate *behaves like a member*. (And indeed, member candidates that don't surpass the calibrated threshold are clearly still members; they just don't have sufficiently high signal to be deemed memorized. This is partly why "membership inference" (Shokri et al., 2017) can be a bit of a confusing term; it's really about whether a sequence "behaves enough like a member" to be reflective of being memorized by the model.) A certified FPR α_N then says how often the same calibrated threshold would incorrectly deem a non-member to be memorized; at that level, some members that clear the threshold are indistinguishable from non-members with respect to their score (up to that certified FPR), and for them we may be over-claiming memorization. The calibrated rate corrects for this (Eq. 8). FWER and FDR, by

Saying something about either of these rates means changing the rule in the conformal test that turns p-values into claims. Recall that our conformal test decides whether to make a claim for each candidate on its own: for a specific candidate $\boldsymbol{s}$, the test claims memorization when that candidate's p-value is at most α_N (Eq. 4), with the cutoff fixed by the controls alone. A **multiple-testing procedure** instead uses the whole family of J p-values to decide which candidates to make a claim about. Both rates depend on the non-member candidates being tested (those indexed by $\mathbb{S}_0$), which (when testing candidates with unknown membership) we don't know. Not knowing $\mathbb{S}_0$ also means that a false positive can't be identified when it occurs. So we ask for a level the rate never exceeds, regardless of which of the J candidates are non-members; a procedure that achieves this bounds the rate at that level. We next discuss procedures that do this for FWER and FDR, respectively.

Bonferroni and FWER. Even though Prop. 1 doesn't *determine* the FWER, it does *say something* about the FWER, coming from the union bound. The union bound says that the probability that any of several events occurs is at most the sum of those events' probabilities. Here, the events are "candidate j is a false positive," one for each $j \in \mathbb{S}_0$; by Prop. 1 each of those events has probability at most α_N. Therefore,

$$\begin{aligned} \mathrm{FWER} &= \Pr\big[\,|\mathbb{M} \cap \mathbb{S}_0| \geq 1\,\big] \\ &\leq \sum_{j \in \mathbb{S}_0} \Pr\big[\,j \in \mathbb{M}\,\big] \;\leq\; J_0\,\alpha_N \;\leq\; J\,\alpha_N, \qquad \text{where } J_0 = |\mathbb{S}_0|. \end{aligned} \tag{27}$$

The tighter of the last two bounds, $J_0\alpha_N$, is valid, but its value is unknown because $\mathbb{S}_0$ (and so J_0) is unknown. The bound we can actually compute is $J\alpha_N$, since we do know J (we chose it). To guarantee $\mathrm{FWER} \leq \delta$ for a target δ chosen in advance (a level for the FWER, playing the analogous role that α does for the FPR), one sets $\alpha_N \leq \delta/J$. This is the Bonferroni correction.[9]

For our memorization setting, the Bonferroni correction isn't practically useful. The reason is that α_N can't be made arbitrarily small: the smallest FPR that N controls can certify is $1/(N+1)$ (Eq. 5). So Bonferroni's requirement of $\alpha_N \leq \delta/J$ can only be met when δ/J is at least $1/(N+1)$, i.e., $1/(N+1) \leq \delta/J$. Put one way, $J \leq \delta(N+1)$: with N controls, Bonferroni can hold the FWER at δ over at most $\delta(N+1)$ candidates. Put another, $N \geq J/\delta - 1$: holding the FWER at δ over J candidates takes at least $J/\delta - 1$ controls. With our $N = 5000$ controls, even a loose $\delta = 5\%$ means we could hold the FWER at 5% over at most 250 candidates (0.05×5001); at $\delta = 0.1\%$, we could hold the FWER over at most 5 candidates. In our experiments, holding the FWER at 0.1% over all $5{,}000$ of our Wikipedia candidates would take about five million controls ($5000/0.001 - 1$).

Benjamini–Hochberg and FDR. For the FDR, a bound at a useful level is available without a correction of this kind, as shown by Bates et al. (2021).[10] The Benjamini–Hochberg (BH) procedure (Benjamini & Hochberg, 1995) sorts the J conformal p-values as $p_{(1)} \leq \cdots \leq p_{(J)}$, finds the largest r with $p_{(r)} \leq r\delta/J$ for a target level δ, and (in the context of our conformal test) would claim memorization for the r candidates with the smallest p-values. It bounds the FDR at δ when the p-values are independent, and also when they're positively dependent in a specific sense: **positive regression dependence on a subset (PRDS)** (Benjamini & Yekutieli, 2001). Bates et al. (2021) prove that marginal conformal p-values computed from one shared calibration set are PRDS, provided the score is continuous[11] and (in our context) the non-member candidates (their inliers) are independent of one another and of the controls (their Theorem 2). So the BH procedure can be run on our p-values unchanged, and bounds the FDR at $\pi_0\delta \leq \delta$, where $\pi_0 = J_0/J$ is the fraction of candidates that are non-members (their Corollary 1).

contrast, aren't being interpreted with respect to this over-claiming: unlike what the FPR measures, they count non-members in the candidate set that clear the threshold (and in our candidate set, there are none).

[9] The same bound covers the FDR, since $\mathrm{FDR} \leq \mathrm{FWER}$ always: the ratio in Eq. 26 is 0 when there is no false positive and at most 1 when there is, so its expectation is at most $\Pr[\,|\mathbb{M} \cap \mathbb{S}_0| \geq 1\,]$.

[10] Bates et al. (2021) study **outlier detection**, so their vocabulary differs from ours: their **calibration set** is our control sample, their **inliers** are our non-members, and their **outliers** are our members. Their marginal conformal p-value (their Eq. 3) is Eq. 4 up to the orientation of the score: they count calibration scores at or *below* the test point's; we count control scores at or *above* the candidate's.

[11] For a score with atoms, as ours has under top-k decoding (§C.4.2), they prove the same for a randomized p-value that breaks ties uniformly at random (their Proposition 4, Appendix A.3). This randomized p-value differs from Eq. 4, where ties count against the candidate.

We don't study either bound in this work. As noted above, Bonferroni isn't practically useful here: holding the FWER over all of our candidates, even at a relatively modest level, would take orders of magnitude more control sequences. BH is something we could study, but it would only apply to extraction procedures with continuous scores, and the results we highlight in the main paper—greedy and top-k decoding—don't use a continuous score (both have atoms, see §C.4.2). Regardless, our main object of study is per-candidate claims about memorization, which is exactly what Prop. 1 is about (and from which we can derive an empirical estimate of calibrated memorization, $\hat{M}$; see Eq. 8).

In future work, it would be perfectly reasonable for others to use the conformal test to make candidate-set-level claims about error rates—for instance, studying a model where it's unknown whether the candidates were in its training data, and where the goal is a list of sequences claimed to be memorized. For BH, one could use the same p-values (Eq. 4), and would need a continuous score and the i.i.d. assumption we already make in practice in our experiments (see §4).

D Population experiments

This appendix gives additional details on the population experiments of §4, which run the conformal test (§3, §C). We first state exactly which confidence interval each plot uses, and why they differ across plots (§D.1). We then describe contamination of the Wikipedia control sample (§D.2): some post-cutoff controls contain pre-cutoff (member) text. The main paper deliberately leaves these contaminants in the control pool, to make the point that contamination pushes the threshold up and so only makes the test more conservative (§C.3). Here, we show what the results would look like with decontamination, which shows how much the calibrated memorization rate rises.

We then extend the experiments in the main paper in three ways. First, suffix length: the main paper uses 10- and 50-token suffixes; here we sweep every length from 1 to 50 tokens (OLMo 2 32B; §D.3). Second, decoding policy: the main paper's greedy and top-k ($k = 40$) policies both use scores with atoms (greedy is binary, and under top-k a sequence can have generation probability exactly 0), so we also run probabilistic extraction under the base distribution at temperature $T = 1$, where the score is continuous (§D.4). Third, model size: under that policy we add OLMo 2 7B and 13B alongside 32B, with a paired comparison of calibrated memorization across sizes (also §D.4).

D.1 Plotted intervals

In the plots in our population experiments (here and in the main paper), all shown intervals are 95% and pointwise: each is a statement about one plotted point, not about a curve as a whole. Where we sweep a threshold (e.g., Fig. 4; the sweep figures below), the shaded band is the pointwise interval at each threshold, drawn continuously. It isn't a simultaneous band over the whole sweep.

There are four kinds of intervals, and which one is appropriate depends on what is random in the specific plot.

- **Clopper–Pearson on a single rate.** For greedy decoding (Fig. 3, Fig. 10a), each sampled sequence is either reproduced verbatim or not, so the member and control counts are binomial and both bars carry exact Clopper–Pearson intervals. For probabilistic extraction, the member rate above a threshold τ is also a binomial proportion, so the member curve in Fig. 4a and in the generation panels of Fig. 13 carries a pointwise Clopper–Pearson band; the certified level α_N is a designed constant and carries no interval (though §C.4.1 shows that α_N is the mean of the Beta distribution of the population FPR α^*, which gives an interval for α^*).
- **Clopper–Pearson shifted by a constant.** The calibrated rate $\hat{M} = \hat{G} - \alpha_N$ (Eq. 8) subtracts a constant from the generation rate $\hat{G}$, so its interval is the Clopper–Pearson interval on $\hat{G}$ translated down by α_N, with the same width. This is the band in Fig. 4b, Fig. 9, Fig. 11, Fig. 12, the absolute rates in Fig. 14a, and the calibrated-rate panels of Fig. 13. The interval is exact because the threshold τ_{α_N} is set by the controls, which are independent of the members: conditional on that threshold, the number of members scoring above it is binomial. So the interval covers one source of randomness, the member sample, holding the threshold fixed at the value the drawn controls set. It doesn't cover the other source, the control draw itself: a different control sample would set a different threshold, with a different population FPR α^*. We keep that second type of

uncertainty out of the band on purpose: the plotted quantity is defined with the certified constant α_N, not the unknown α^*; an interval that also carried the control-draw uncertainty would be an interval for $\hat{G} - \alpha^*$, which is a different quantity. §C.4.3 discusses that second source of uncertainty.

- **Newcombe's independent-proportions interval.** In two figures, the quantity subtracted from the member rate depends on the control draw, so it carries sampling variance of its own. For greedy decoding, the score is binary, so there's no target FPR to choose: the false-positive floor is set by how many controls the model reproduces, and a different control draw would give a different floor (Fig. 10b). For the budget plots at $n = 10^5$ (Fig. 19b), the subtracted term is the measured non-member rate at that budget, again an empirical proportion over the control sample. For a difference of two estimated proportions from independent samples, Newcombe's independent-proportions interval is a useful choice: a Wilson score interval for each rate, combined into one interval for the difference (Newcombe, 1998b).
- **Newcombe's paired-proportions interval.** For the same candidates scored by two models (Fig. 14b), the two rates aren't independent, because every candidate contributes to both. We use Newcombe's paired-proportion interval, which builds a Wilson interval for each model's rate and combines them for the difference with a correction for the correlation between the two rates (Newcombe, 1998a). That correlation is estimated from how often the two models agree on whether a given candidate is claimed memorized. §D.4 explains why this is useful in further detail.

D.2 Score distributions and contamination

Probabilistic extraction scores each suffix by a continuous quantity, so the conformal threshold can be swept across operating points. At 10 tokens, many non-member continuations are highly predictable. Holding the target FPR to 0.1% therefore forces the threshold close to one: the test deems a candidate memorized only when the model generates it with near certainty. At 50 tokens, non-members are almost never generated verbatim, so the threshold giving the same target FPR is orders of magnitude smaller.

It turns out that some of the controls contain pre-cutoff (member) text that survived our post-cutoff filter: the 2 highest-scoring controls, and 10 more we found through an exhaustive, agent-enabled manual check. Ten of these twelve sit at the top of the control distribution, where they raise the extreme-tail threshold; the remaining two score exactly 0, so they don't move it. Fig. 9 shows how removing these 12 verified contaminants changes the estimated calibrated rate. The budget results in §F.2, and the $T = 1$ experiments, including the model-scaling comparison (§D.4), calibrate on this decontaminated pool, which gives $\alpha_N = 4/4989 \approx 0.0802\%$. The results in the main paper and the suffix-length sweep in §D.3 use the full pool ($N = 5000$), for which $\alpha_N = 5/5001 \approx 0.09998\%$. (The main results report the control construction and decontamination status explicitly rather than treating a post-cutoff date as sufficient evidence of non-membership.)

D.3 Varying suffix length

The main paper compares 10- and 50-token suffixes; here we sweep every suffix length from 1 to 50 tokens for OLMo 2 32B, scoring the same 5,000 members and 5,000 matched controls, with the same 50-token prefixes, at every suffix length. Throughout the sweep we use the full control pool ($N = 5000$) rather than the decontaminated one (§D.2). Contamination isn't easily defined at short suffixes: at one or two tokens, nearly every control continuation likely appears somewhere in pre-cutoff text, so removing "contaminated" controls would remove the whole pool.

Greedy decoding has no threshold to calibrate, so Fig. 10 shows the two raw rates and their difference. Both fall with length; the non-member rate falls faster, and the difference peaks near $\ell = 3$ before declining. Fig. 3 in the main paper reports two non-member quantities: the control bars give the in-sample rate, and the dashed line gives the certified FPR $\alpha_N = (m + 1)/(N + 1)$, where m is the number of controls the model reproduces. Fig. 10b subtracts the in-sample rate, which differs from α_N by less than $1/(N+1)$. Subtracting an estimated rate is why Fig. 10b carries a Newcombe interval rather than a shifted one (§D.1). At the peak at $\ell = 3$, the certified FPR is about 22%, so the difference there is between two enormous rates; each point on the curve is its own operating point, at its own FPR, so heights aren't comparable across lengths.

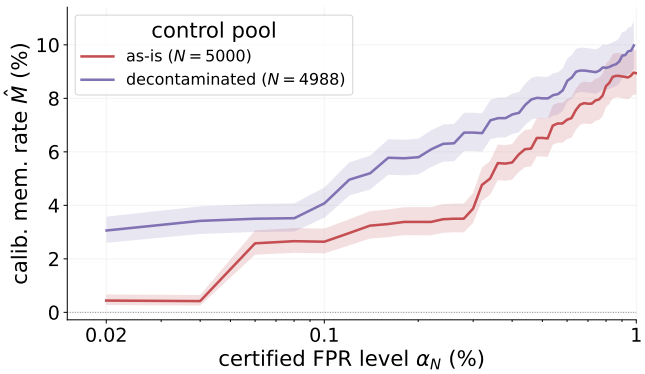
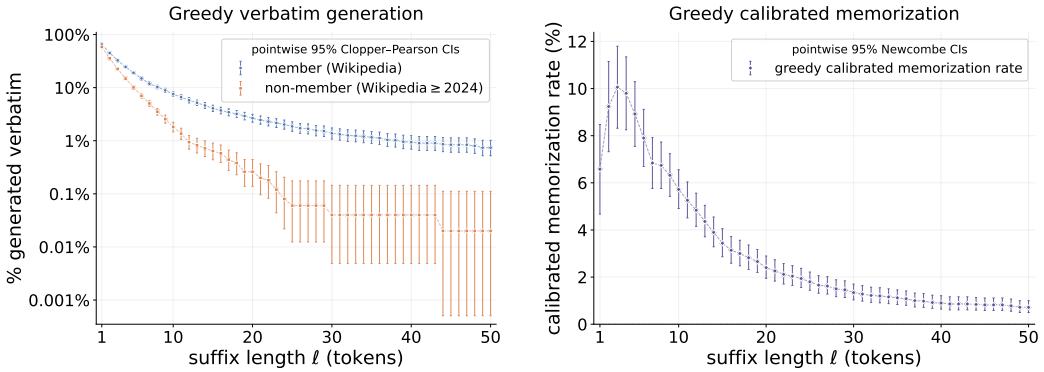


Figure 9: **Effect of decontaminating the matched Wikipedia controls.** We zoom in on the 50-token, sub-1% FPR regime, comparing the as-is control pool ($N = 5000$) against the decontaminated pool ($N = 4988$). Decontamination drops 12 sequences: the 2 the model generates with probability above 10^{-3}, and 10 more found through an exhaustive, agent-enabled manual check. Contaminated controls sit at the top of the control distribution, so they raise the extreme-tail threshold and suppress the calibrated rate. Removing contaminated controls raises the calibrated rate from 0.44% to 3.06% at the tightest certified level ($\alpha_N \approx 0.02\%$). Bands are 95% Clopper–Pearson intervals on the member rate, shifted down by the constant α_N (§D.1).

Probabilistic extraction behaves differently, because its score is continuous: at each length we can set the threshold to calibrate to a target FPR. Fig. 11 plots the member generation rate above the length-specific calibrated threshold at three targets ($\alpha = 0.1\%, 1\%, 10\%$). The member rate rises with length: at $\alpha = 0.1\%$ it climbs from about 0.5% at $\ell = 1$ to about 2.7% at $\ell = 50$, and the curves for larger target FPRs (1%, 10%) sit higher throughout. This is the opposite behavior from greedy: as ℓ grows, non-member scores concentrate near zero, so the threshold that certifies a given FPR falls, and more members clear it. Fig. 12 shows the corresponding calibrated rates $\hat{M} = \hat{G} - \alpha_N$; since α_N is a constant at each target FPR, these are the member curves shifted down by α_N. At $\alpha = 0.1\%$ the calibrated rate rises to about 2.64 percentage points by $\ell = 50$ (the 2.64% of §4); at $\alpha = 1\%$ to about 9%; at $\alpha = 10\%$ it rises to about 18% by $\ell \approx 25$ and then levels off. So a longer suffix certifies more memorization at a matched FPR even though its raw generation rate is lower, because at a common threshold the non-member rate collapses faster than the member rate.



(a) Member and non-member generation rates, with pointwise 95% Clopper–Pearson intervals.

(b) Member rate minus the in-sample non-member rate, with pointwise 95% Newcombe independent-proportions intervals.

Figure 10: **Greedy discoverable extraction across suffix lengths (OLMo 2 32B).** Both rates fall with length, and the non-member rate falls faster: at $\ell = 1$ it is 58% against 65% for members, and at $\ell = 50$ it is 0.02% against 0.74%. The difference in (**b**) therefore rises first, as the gap opens over the first few tokens, peaking near $\ell = 3$ at $\approx$10 percentage points, and then falls as both rates shrink, to 0.72 points at $\ell = 50$.

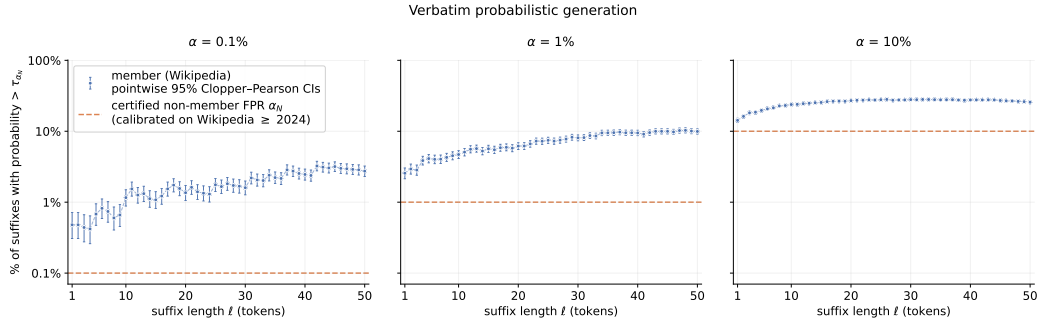


Figure 11: **Probabilistic member generation across suffix lengths at a fixed target FPR (OLMo 2 32B, full control pool).** For each target FPR ($\alpha = 0.1\%, 1\%, 10\%$), the member rate is the share of members scoring above the threshold calibrated for that length; the dashed line is the certified FPR α_N. Whiskers are pointwise 95% Clopper–Pearson intervals.

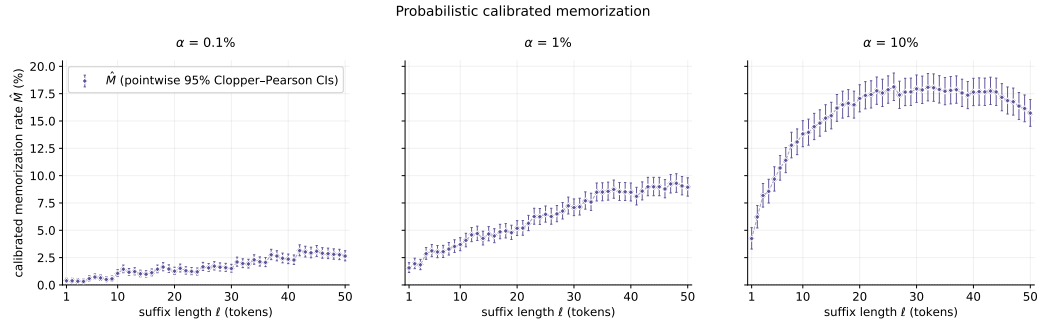


Figure 12: **Probabilistic calibrated memorization across suffix lengths (OLMo 2 32B, full control pool).** For each target FPR ($\alpha = 0.1\%, 1\%, 10\%$), the calibrated rate $\hat{M} = \hat{G} - \alpha_N$ (Eq. 8) is the member rate above the threshold calibrated for that length, minus the certified FPR α_N. Whiskers are pointwise 95% Clopper–Pearson intervals on $\hat{G}$, shifted down by α_N.

D.4 TESTING A DIFFERENT DECODING POLICY

The main paper's results use decoding policies that produce scores with atoms. Greedy's score is binary; under top-k a verbatim candidate might not be reachable from expanding the top-k tokens at every step, and in this case has generation probability exactly 0.

Atoms have two consequences for calibration and probabilistic extraction. First, the certified level has a ceiling: the largest attainable α_N is the share of controls with a non-zero score, which at 50 tokens under top-40 is 10.4% for 32B (the "max α_N" annotation in Fig. 4b), 8.0% for 13B, and 6.4% for 7B. Second, the Beta law for the population FPR (§C.4.1) assumes a continuous score; §C.4.2 extends it to every level below the ceiling $\alpha^\dagger$. (This works for top-k because the atom is the block of zero scores, which sits at the very bottom of the score range, while the calibrated threshold is one of the highest control scores. As long as at least K controls score above zero, the threshold lands clear of the block.)

To include a decoding policy with no atoms, we repeat the experiment under the base policy, sampling from the full softmax at temperature $T = 1$. We do this for OLMo 2 7B, 13B, and 32B, at 10 and 50 tokens on the same $5{,}000$ members, calibrating on the decontaminated control pool ($N = 4988$), so $\alpha_N = 4/4989 \approx 0.080\%$ at the 0.1% target FPR. Under this policy, every sequence has positive generation probability, so there's no certified FPR ceiling, the score is continuous, and the Beta law applies directly at every certified level.

Fig. 13 shows the results. The conclusions of §4 hold under this policy for every model size: the 50-token-suffix calibrated rate exceeds the 10-token-suffix rate across the whole sweep. At $\alpha_N \approx 0.08\%$, the 50-token calibrated rates are 2.74%, 3.68%, and 3.70% for 7B, 13B, and 32B, compared to 1.16%, 1.74%, and 1.18% at 10 tokens.

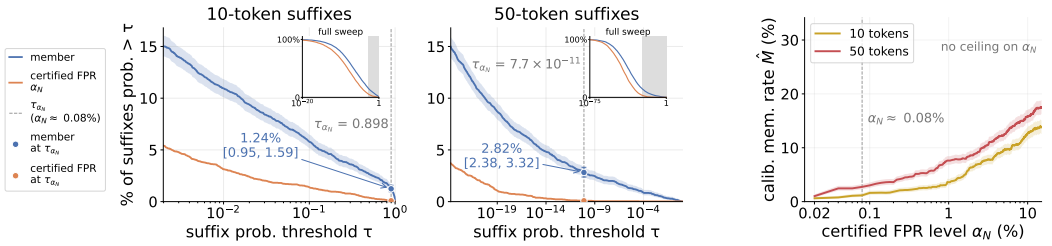
Model scaling. We also compare model sizes under the $T = 1$ policy (Fig. 14). The same $J = 5{,}000$ candidates are scored for 7B, 13B, and 32B. Write $\hat{M}$ and $\hat{M}'$ for the calibrated memorization rates of two model sizes, with scores $\mathsf{score}, \mathsf{score}'$ and thresholds $\tau_{\alpha_N}, \tau'_{\alpha_N}$. Both sizes calibrate on the same N controls at the same target FPR, so they share the same α_N. Therefore,

$$\begin{aligned}\hat{M} - \hat{M}' &= (\hat{G} - \alpha_N) - (\hat{G}' - \alpha_N) \\ &= \hat{G} - \hat{G}' \\ &= \frac{1}{J}\sum_{j=1}^{J}\Big(\mathbf{1}\{\mathsf{score}(\boldsymbol{s}_j) > \tau_{\alpha_N}\} - \mathbf{1}\{\mathsf{score}'(\boldsymbol{s}_j) > \tau'_{\alpha_N}\}\Big). \end{aligned} \tag{28}$$

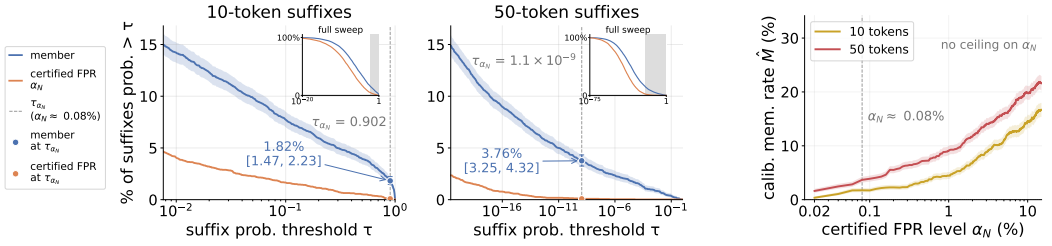
In the last row, each summand is $+1$, -1, or 0; it's 0 exactly when the two sizes agree about candidate j. So the difference is determined by the candidates the two sizes disagree on with respect to claiming memorization, and the band in Fig. 14b is an interval for that paired difference, $\hat{M} - \hat{M}'$ (Eq. 28). The pairing therefore distinguishes a scale effect in memorization from the fact that larger models tend to be better predictors.

The calibrated rate rises with model size at every target FPR: at $\alpha = 0.1\%$ it is 2.74%, 3.68%, and 3.70% for 7B, 13B, and 32B; at 1%, 7.62%, 9.08%, and 11.38%; at 10%, 15.78%, 19.82%, and 23.22% (Fig. 14a). Fig. 14b plots the paired difference $\hat{M} - \hat{M}'$ (Eq. 28) for the three size pairs, sweeping the target FPR from 0.1% to 20%. In each pair, $\hat{M}$ is the calibrated rate of the larger size and $\hat{M}'$ that of the smaller one. A positive value at a given level therefore means the larger size claims memorization for more of the shared candidates than the smaller one does, and the value is how many percentage points more. We show pointwise 95% Newcombe paired-proportion intervals (§D.1). At a given target FPR, if the interval lies entirely above zero, every difference it covers is positive, so the larger size's calibrated rate exceeds the smaller one's by at least the interval's lower endpoint. If it includes zero, then the results don't resolve whether one of the two sizes has a higher calibrated memorization rate.

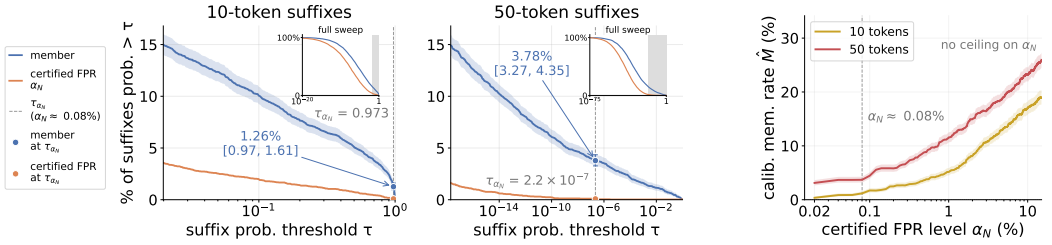
The paired differences are positive across that whole range for all three pairs. Two of the three pairs have intervals entirely above zero at every one of those target FPRs: 13B minus 7B ($+0.94$ points $[0.64, 1.27]$ at 0.1%, $+4.04$ $[3.34, 4.74]$ at 10%) and 32B minus 7B ($+0.96$ $[0.57, 1.37]$ and $+7.44$ $[6.60, 8.28]$). The third, 32B minus 13B, has an interval above zero at every target FPR from 0.33%



(a) 7B: verbatim generation (same prefixes, 10- and 50-token suffixes) (b) 7B: calibrated memorization



(c) 13B: verbatim generation (same prefixes, 10- and 50-token suffixes) (d) 13B: calibrated memorization



(e) 32B: verbatim generation (same prefixes, 10- and 50-token suffixes) (f) 32B: calibrated memorization

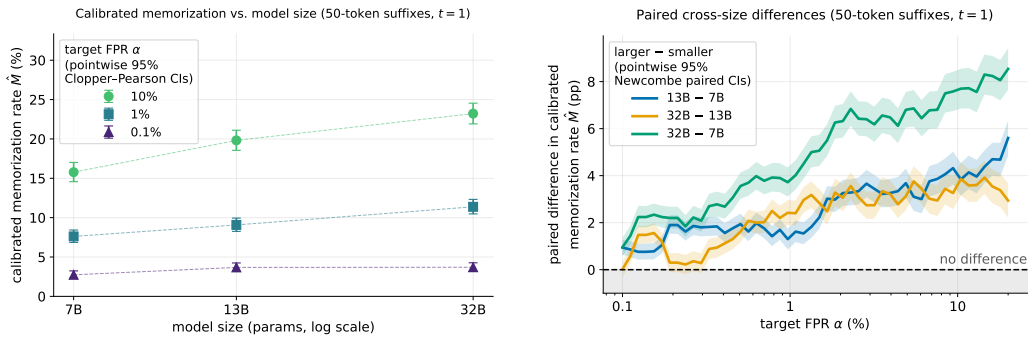
Figure 13: **Generation and calibrated memorization rate for OLMo 2 base models on Wikipedia ($T = 1$).** We repeat the experiment shown in Fig. 4 for OLMo 2 7B, 13B, and 32B and, instead of top-k decoding ($k = 40$, $T = 1$), we use the base policy ($T = 1$). With this decoding policy, no candidate or control sequences have 0% generation probability, so there is no ceiling on α_N. The top row is 7B, middle row is 13B, and the bottom row is 32B. (All six panels calibrate on the decontaminated control pool, $N = 4988$; see Fig. 9. This is why $\alpha_N = 4/4989 \approx 0.0802\%$ here rather than the $\approx 0.1\%$ of Fig. 4.) (**a**), (**c**), (**e**) Complementary CDFs of scores: a point (τ, y) means $y\%$ of sequences score above τ (i.e., $q_{\theta,\phi}(\boldsymbol{s} \mid \boldsymbol{p}) > \tau$). Each panel is cropped to highlight the $\alpha_N \approx 0.08\%$ operating point, beginning where the member rate falls to 15%. We visualize the three models this way because the plotted sweep spans roughly twenty-five orders of magnitude at 10 tokens and a hundred at 50, which leaves the threshold region unreadably compressed. The inset shows that full sweep, with the cropped window shaded. (**b**), (**d**), (**f**) Calibrated memorization (Eq. 8) for 10- and 50-token suffixes across certified FPR levels α_N. The 50-token rate is the larger of the two across the sweep. Bands are 95% Clopper–Pearson intervals on the member rate, shifted by the constant α_N in (**b**), (**d**), (**f**).

onwards ($+3.40$ $[2.71, 4.10]$ at 10%), but not at the tightest ones: at 0.1% it is $+0.02$ $[-0.31, 0.35]$, with 13B and 32B at the same 3.7%.

D.5 Margin

The calibrated threshold makes a binary claim, but the strength of evidence above it varies. We can quantify this using a notion of **margin**: how many orders of magnitude the candidate's score is above a threshold. For a candidate $\boldsymbol{s}$ and a threshold τ, define its margin in orders of magnitude by

$$\Delta(\boldsymbol{s}; \tau) = \log_{10} \frac{\mathsf{score}(\boldsymbol{s})}{\tau}. \tag{29}$$



(a) Absolute calibrated rates

(b) Paired cross-size differences

Figure 14: **Calibrated verbatim memorization versus OLMo 2 model size ($T = 1$).** The same 50-token Wikipedia candidates and decontaminated controls ($N = 4988$) are scored for the 7B, 13B, and 32B base models under the base policy ($T = 1$). Left: absolute calibrated rates at three target FPRs, with pointwise 95% Clopper–Pearson intervals shifted down by α_N. Right: paired differences across the sweep of target FPRs, with pointwise Newcombe paired-proportion intervals (Newcombe, 1998a); an interval above zero supports a larger calibrated rate for the larger model.

Positive Δ means s clears τ, and larger values place it farther above τ. Setting $\tau = \tau_{\alpha_N}$ gives the margin above the calibrated threshold.

Two candidates in the same experiment are measured against the same threshold, so the one with the larger margin is the one with the higher score. A large margin means the candidate's score sits orders of magnitude above what the matched non-members reached, which we can interpret as stronger evidence that the model memorized it. A margin just above zero is the weakest evidence the test can claim on: the candidate clears the threshold, but it scores about as well as the control that set the threshold (and would be among the first to no longer be claimed memorized if a tighter α_N was used to set a different threshold).

We can look directly at the scores to see which claimed candidates fall in the range where the controls that set the threshold also land (Fig. 4a). Those are the claims that risk being false positives: the share of the member generation rate that the certified FPR accounts for (§4). Margin is just another lens on this, but one that is normalized: dividing by the threshold sets it to zero, so a margin of 2 means two orders of magnitude above the threshold whether that threshold is 8.9×10^{-6} at 50 tokens or 0.95 at 10 (§4). (§5 does this across books, comparing coverage five orders of magnitude above each book's own threshold.)

Greedy decoding has no margin to report, since its score is 0 or 1. Every candidate that clears the threshold has the same score (§C.4.4).

E CENSUS EXPERIMENTS

We run census experiments on the four matched pairs of books shown in Tab. 1. For each author, we include a book known to be in Llama 3.1's training data from the Books3 corpus and a book published after Llama 3.1's training cutoff. For Dan Brown and Suzanne Collins, the respective pairs of books are from the same series. We score every window with the sliding-window verbatim probabilistic extraction procedure of Cooper et al. (2026a), under top-k decoding with $k = 40$. We use this policy because it's the one that prior work on books uses. Because every window of each book is scored, the coverage we report is the exact value for that book rather than an estimate, so none of our census results carries an interval (contrast the plotted intervals in §D.1).

Table 1: **Matched book pairs.** Llama 3.1's pretraining cutoff is December 2023. Each Books3 title is paired with a same-author book published after the cutoff.

Author	Member (In-training from Books3)	Non-member (Post-cutoff control)
Dan Brown	*The Da Vinci Code* (Brown, 2003)	*The Secret of Secrets* (Brown, 2025)
Suzanne Collins	*The Hunger Games* (Collins, 2008)	*Sunrise on the Reaping* (Collins, 2025)
Ta-Nehisi Coates	*We Were Eight Years in Power* (Coates, 2017)	*The Message* (Coates, 2024)
Joseph Finder	*Killer Instinct* (Finder, 2006)	*The Oligarch's Daughter* (Finder, 2025)

E.1 CALIBRATING THRESHOLDS AND DECONTAMINATION

For a book pair, the calibrated threshold is the largest score among decontaminated windows of the post-cutoff book, rounded up to the next power of ten. That is, denote a control book's set of sliding-window sequences to be control. Then

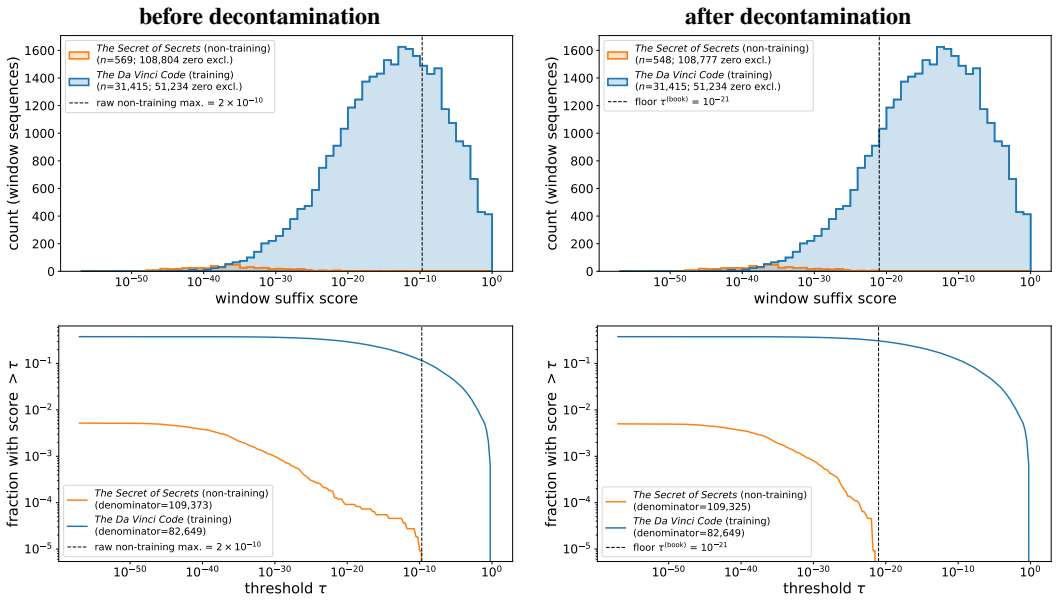
$$\tau^{(\text{book})} = 10^{\lceil \log_{10} \max_{w' \in \text{control}} \mathsf{score}(w') \rceil}. \tag{30}$$

Rounding upward is conservative. It avoids attaching meaning to an unreliable floating-point mantissa at probabilities near 10^{-20} and below.

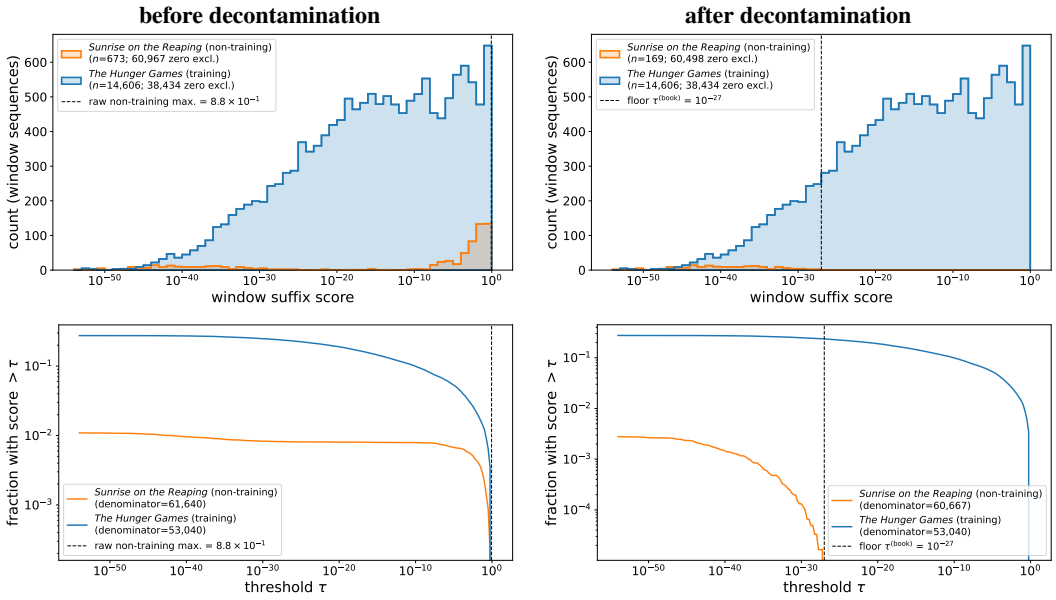
As part of calibrating the threshold, we need to decontaminate the matched control books. This is because, even when a book post-dates training, individual passages may quote training data. For decontamination, we inspect the high-scoring control tail, decode the underlying text, and remove only passages verified as being reused from earlier sources; the in-training book is never culled. We developed a rigorous, deliberately rote process for culling, in which we used Claude to perform verified searches on the internet to identify provenance of the high-scoring windows. The process was multi-turn and strictly sequential: rather than assessing all candidate passages at once, each was worked through in its own step. Every single identified passage was manually verified by a person before culling.

For *Sunrise on the Reaping*, for example, the removed passages include repeated lines from Edgar Allan Poe's "The Raven" and quotations from *The Hunger Games*. We use each control book's highest-scoring own-prose passage to set the threshold (Eq. 30).

Fig. 15 shows how the non-member distributions shift after decontamination for Dan Brown and Suzanne Collins; Fig. 16 shows the same results for Ta-Nehisi Coates and Joseph Finder.

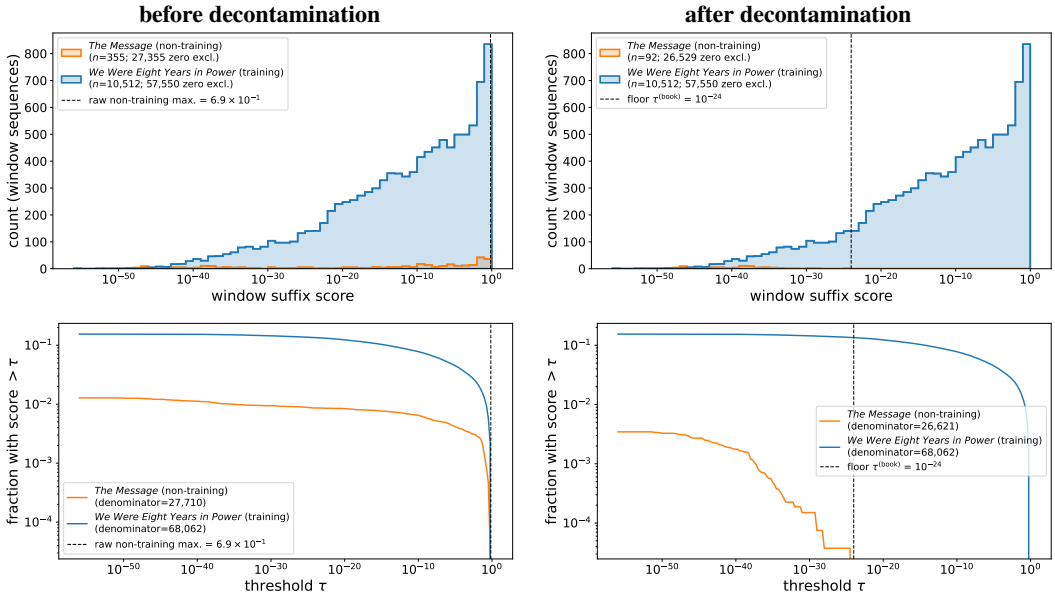


(a) Dan Brown: *The Secret of Secrets* (non-member) and *The Da Vinci Code* (member).

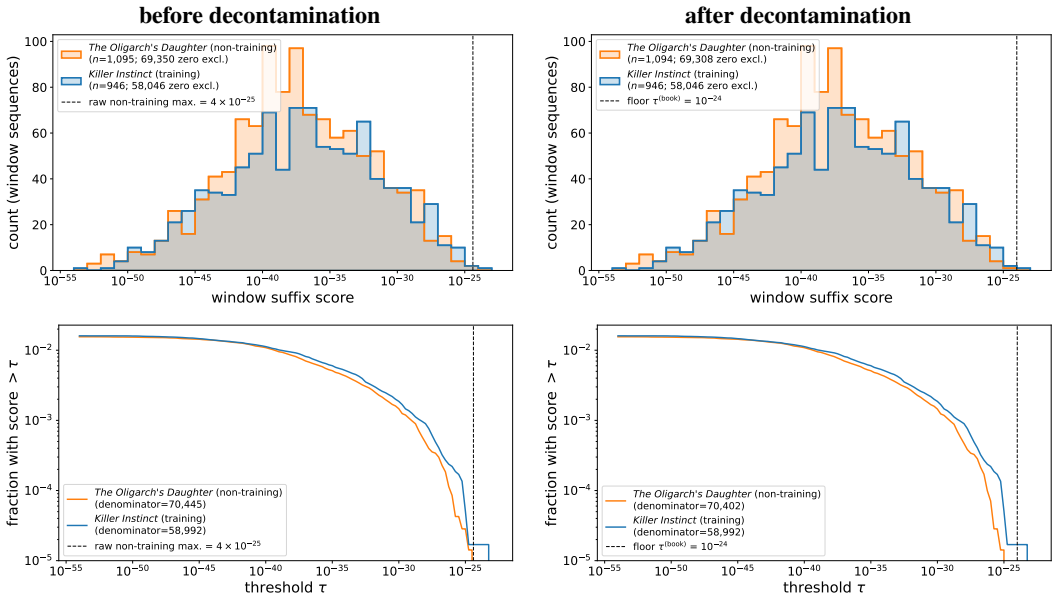


(b) Suzanne Collins: *Sunrise on the Reaping* (non-member) and *The Hunger Games* (member).

Figure 15: **Matched-control decontamination for the Brown and Collins pairs.** The top row within each pair is the score histogram and the bottom row the complementary CDF, before and after verified reused passages are removed.



(a) Ta-Nehisi Coates: *The Message* (non-member) and *We Were Eight Years in Power* (member).



(b) Joseph Finder: *The Oligarch's Daughter* (non-member) and *Killer Instinct* (member).

Figure 16: **Matched-control decontamination for the Coates and Finder pairs.** The top row within each pair is the score histogram and the bottom row the complementary CDF, before and after verified reused passages are removed. Note that the member and non-member distributions are indistinguishable for Finder books.

E.2 Additional results

Where the signal occurs and how far it clears the floor. Beyond reporting coverage above the threshold, following Cooper et al. (2026a), we can plot where memorization occurs within a book using a heatmap. Fig. 17 shows an example of this type of result for *The Hunger Games* and Llama 3.1 70B, using verbatim probabilistic extraction, where the minimum probability that counts as extraction is just above the calibrated threshold.

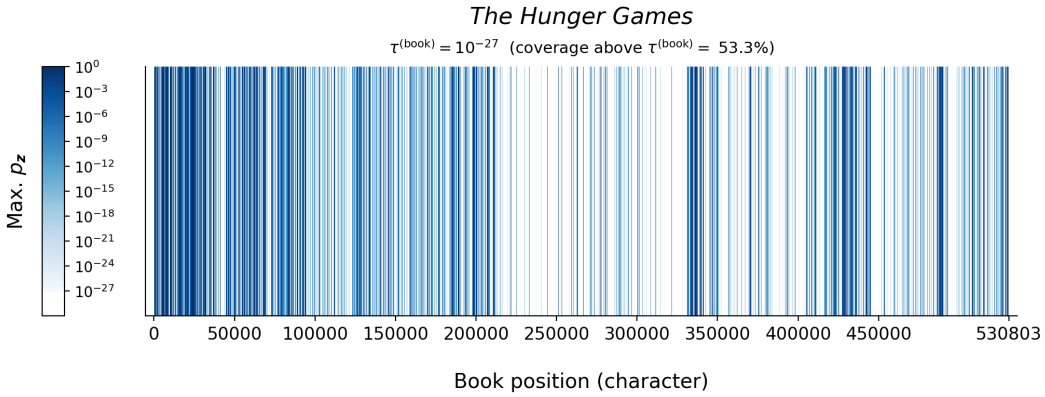
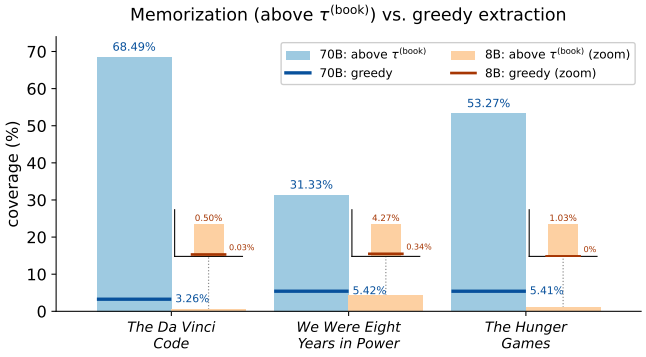


Figure 17: **Memorization across *The Hunger Games*.** Each character is colored by the largest 50-token suffix score among the overlapping windows whose suffix covers it. The minimum probability that counts as evidence of memorization comes from calibrating against the matched control book, *Sunrise on the Reaping*.

Margin. In the main paper, we report results using coverage CCDFs. This is because the calibrated threshold makes a binary claim, but the strength of evidence above it varies. That way, we can visualize how much coverage there is above the calibrated threshold, as well as how coverage changes as we increase probability above that threshold. We quantify this with the margin (Eq. 29), instantiated at each book's own calibrated threshold:

$$\Delta(\boldsymbol{s}; \tau^{(\text{book})}) = \log_{10} \frac{\mathsf{score}(\boldsymbol{s})}{\tau^{(\text{book})}}. \tag{31}$$

Positive Δ means $\boldsymbol{s}$ clears the threshold; larger values place it farther from the strongest matched non-member that sets the calibrated threshold. Fig. 5 in the main paper marks the coverage five orders of magnitude above each book's own $\tau^{(\text{book})}$, which is coverage at $\Delta = 5$.



Training book	$\tau^{(\text{book})}$	
	70B	**8B**
Da Vinci Code	10^{-21}	10^{-22}
Eight Years	10^{-24}	10^{-36}
Hunger Games	10^{-27}	10^{-31}
Killer Instinct	10^{-24}	10^{-26}

Figure 18: **Single-document census results.** For each book, we plot coverage for Llama 3.1 70B and 8B. The bars show probabilistic extraction coverage above the respective calibrated threshold for each pair. We use lines on the bars to show greedy coverage for the same book and model. *Killer Instinct* is not plotted on the left because it carries effectively no memorization signal.

F EXTRACTABILITY AT A FINITE QUERY BUDGET

This appendix provides additional details on experiments concerning extractable memorization (§6). To start, we fully state the derivation of the sampling-budget accounting.

F.1 QUERY BUDGET

Fix a prompt $\boldsymbol{p}$ and let

$$p_{\boldsymbol{s}} \coloneqq q_{\theta,\phi}(\boldsymbol{s} \mid \boldsymbol{p})$$

be the per-query generation probability. Across n independent samples, the probability of generating $\boldsymbol{s}$ at least once is

$$\Pr_{\text{gen}}(n) = 1 - (1 - p_{\boldsymbol{s}})^n. \tag{32}$$

For a target success probability $c = 1 - \delta \in (0, 1)$, the required integer budget is

$$n(p_{\boldsymbol{s}}; c) = \left\lceil \frac{\log(1-c)}{\log(1-p_{\boldsymbol{s}})} \right\rceil. \tag{33}$$

Conversely, the smallest per-query probability that reaches confidence c within a fixed budget n is

$$p_{\boldsymbol{s}}(n; c) = 1 - (1-c)^{1/n}. \tag{34}$$

For small $p_{\boldsymbol{s}}$, $\log(1 - p_{\boldsymbol{s}}) \approx -p_{\boldsymbol{s}}$, giving

$$n(p_{\boldsymbol{s}}; c) \approx \frac{-\log(1-c)}{p_{\boldsymbol{s}}}.$$

At $c = 0.95$, the numerator is approximately 2.996, so reaching 95% confidence costs roughly three times the inverse per-query probability. This is where the 3×10^{21} samples number comes from in our discussion in §6.

In the remainder of this appendix, we provide budget views (analogous to Fig. 6) for both population conformal-test experiments and census experiments. In all, we show results for $c = 95\%$ (i.e., $\delta = 5\%$). We then show additional results on deterministic realistic extraction of low-probability memorized sequences.

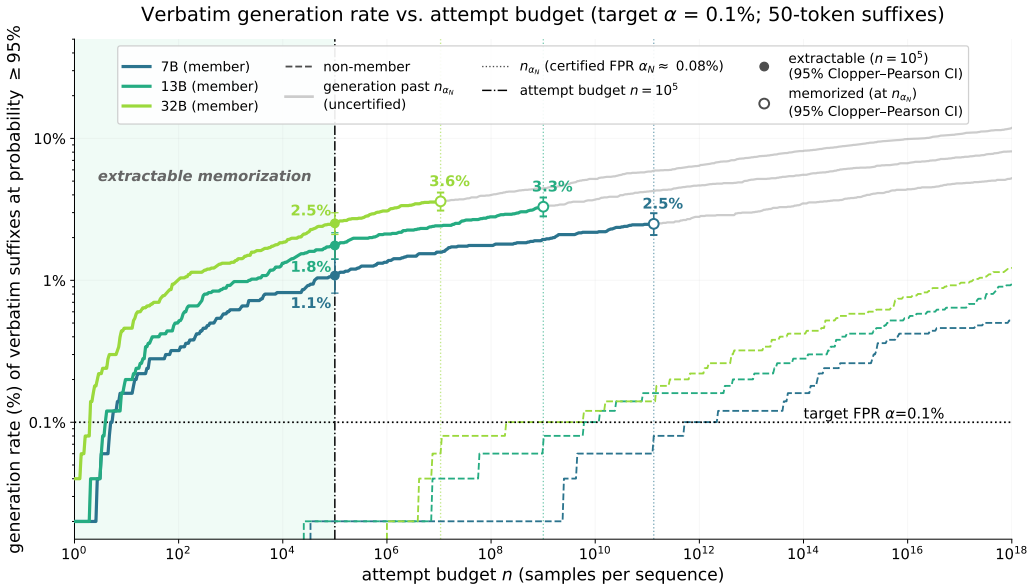
F.2 POPULATION BUDGET VIEWS

We show two combined budget views for OLMo 2 models on Wikipedia. Both plots show rates at $1 - \delta = c = 95\%$.

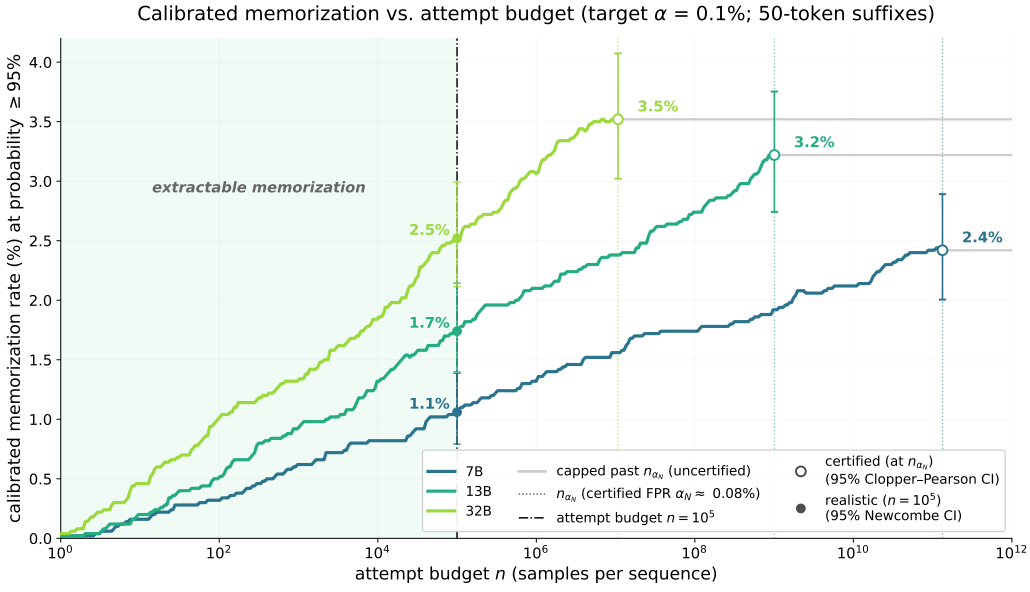
In Fig. 19a, we show the budget view for 7B, 13B, and 32B for the raw member and non-member generation rates. The y-axis shows these rates, and so we can treat the non-member generation rate as the FPR. We indicate a reference line for where the budget crosses a target FPR of $\alpha = 0.1\%$; the level the calibrated threshold actually certifies is $\alpha_N = 4/4989 \approx 0.0802\%$, just below that target α. These budget figures calibrate on the decontaminated control pool ($N = 4988$; Fig. 9), which gives $K = \lfloor 0.001 \times 4989 \rfloor = 4$. The results in the main paper use the full pool ($N = 5000$), for which $\alpha_N = 5/5001 \approx 0.09998\%$. The mint-green shaded region marks budgets up to $n = 10^5$, our illustrative realistic budget. For each model, the hollow dot marks the member generation rate at $n_{\alpha_N} = \lceil \log(0.05)/\log(1 - \tau_{\alpha_N}) \rceil$ (Eq. 33): the budget at which a sequence whose per-query probability equals the calibrated threshold is generated at least once with probability 95%. The filled dot marks the rate at the realistic extraction budget $n = 10^5$. Both dots carry exact Clopper–Pearson intervals. At a fixed budget, whether a sequence clears that budget is determined by its own per-query probability, so it's not a random event; the only randomness is which sequences we sampled. The count of sampled sequences that clear it is therefore binomial, and the Clopper–Pearson interval is exact for binomial proportions.

In Fig. 19b, we show the corresponding calibrated memorization rates (Eq. 8) for the results in Fig. 19a, where both rates count only sequences that score above τ_{α_N}. Note that the two sets of annotated dots have different intervals because they subtract different objects. At n_{α_N}, the subtracted term is the certified level α_N; this is a constant that carries no sampling variance, so the interval is the member Clopper–Pearson interval shifted down by α_N (the same width, just translated down).

In contrast, at $n = 10^5$, the subtracted amount is the measured non-member rate at that budget. That rate is an empirical proportion over a finite control sample, so a different draw of controls would have produced a different value. Both terms in the difference therefore carry sampling variance, so the interval has to account for both. Newcombe's independent-proportions interval takes a Wilson score interval for each rate and combines them into a single interval for the difference.

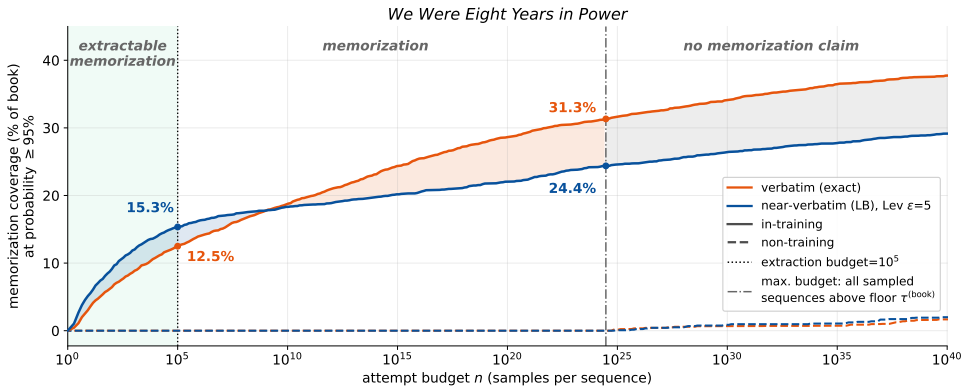


(a) Member and non-member generation rates reachable at a given budget (certified $\alpha_N \approx 0.0802\%$)

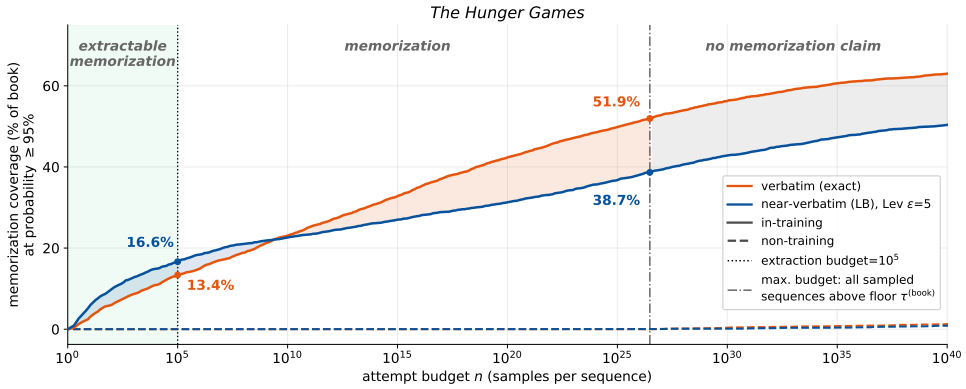


(b) Calibrated memorization rate reachable at a given budget, at target FPR $\alpha = 0.1\%$ (certified $\alpha_N \approx 0.0802\%$)

Figure 19: **Two budget-level views of OLMo 2 on Wikipedia.** Both plots use 50-token prefixes, 50-token suffixes, and require at least 95% probability of one successful generation within the stated number of independent samples.



(a) *We Were Eight Years in Power*; verbatim and near-verbatim use $\tau^{(\text{book})} = 10^{-24}$.



(b) *The Hunger Games*; the plot uses the more conservative near-verbatim $\tau^{(\text{book})} = 10^{-26}$.

Figure 20: **Coverage against attempt budget for two additional Llama 3.1 70B book censuses.** The non-member reference curves rise above zero only after the budget associated with the matched threshold, $\tau^{(\text{book})}$. We calibrate $\tau^{(\text{book})}$ separately for each extraction procedure, so it can produce different results for verbatim and near-verbatim for the same pair of books.

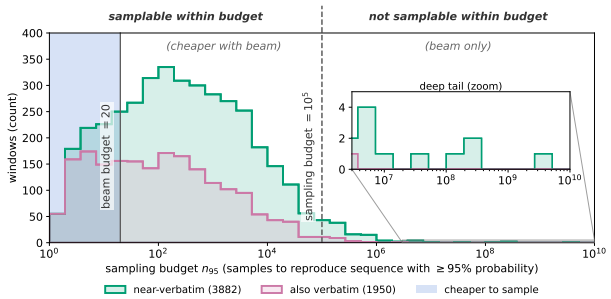
F.3 Census budget views

We provide results for *We Were Eight Years in Power* and *The Hunger Games* in Fig. 20. These complement the results for *The Da Vinci Code* in Fig. 6 We again omit results for *Killer Instinct*, since we detect effectively no memorization in our experiments for that book. For near-verbatim probabilistic extraction, we use the same sliding-window procedure to produce prompts and candidates and the same decoding policy (top-k, $k = 40$) as for the verbatim results; what differs is the scoring, which uses the efficient lower-bound algorithm from Cooper et al. (2026b), with Levenshtein-distance pruning enabled with $\varepsilon = 5$.

F.4 DETERMINISTIC RECOVERY

Random sampling is not the only way to decode outputs. Cooper et al. (2026b) use a beam-search-based algorithm to return a deterministic pool of 800 continuations per candidate, with cost comparable to approximately 20 samples. For these experiments, we run the baseline algorithm, which performs no pruning to guide the beam search to within the ε-ball of the candidate sequence, and we post-process the results to filter to those within Levenshtein distance of 5 tokens from the candidate. That is, we check the returned pool for verbatim and near-verbatim recovery, then compare the recovered sequences with what can be sampled inside the illustrative budget.

Fig. 21 shows how many recovered windows sit at each sampling budget. The x-axis is n_{95}, the number of random samples needed to reproduce a sequence with probability 95%, and the y-axis is the count. The near-verbatim results show every window the algorithm recovered within Levenshtein distance of 5, and the verbatim results are the subset that also recovered the exact target (both series are plotted at the near-verbatim probability, so verbatim nests inside near-verbatim in every bin). Counts are per sliding window rather than per distinct passage. The gray-blue region marks budgets below the beam's own cost of roughly 20 passes, where random sampling is the cheaper way to obtain a sequence. Fig. 22 provides an alternate view of the same results, limited to near-verbatim extraction. Each dot is one recovered window, placed at its extraction probability and the sampling budget n_{95} that probability implies; the flat line is the beam's fixed cost, so the vertical drop from a dot down to that line is how much cheaper the deterministic procedure was at extraction.



	Coverage (%)	
Recovered in	**verbatim**	**near-verbatim**
top-1 decode	6.1	13.6
top-20 beams	10.1	16.9
top-800 pool	10.4	17.1
not samplable within 10^5	1.8	2.3

Figure 21: **Deterministic recovery for *The Da Vinci Code*.** The near-verbatim lower-bound algorithm surfaces sequences within a computational budget comparable to roughly 20 independent samples; some of the recovered memorized content wouldn't be sampled within 10^5 attempts. Verbatim recovery requires the exact target, while near-verbatim requires only a sequence within Levenshtein distance of 5 tokens of it (near-verbatim recovery subsumes verbatim recovery). "Cheaper to sample" means the sequence's probability is high enough that random sampling reaches it in fewer draws than the roughly 20 passes the beam-search algorithm costs. The table reports coverage for different subsets of the pool of sequences returned by the algorithm. For instance, we get 6.1% verbatim coverage from looking only at the top-1 sequence returned in the 800-sequence pool for each candidate. The last row shows the coverage that sampling would not reach within 10^5 attempts.

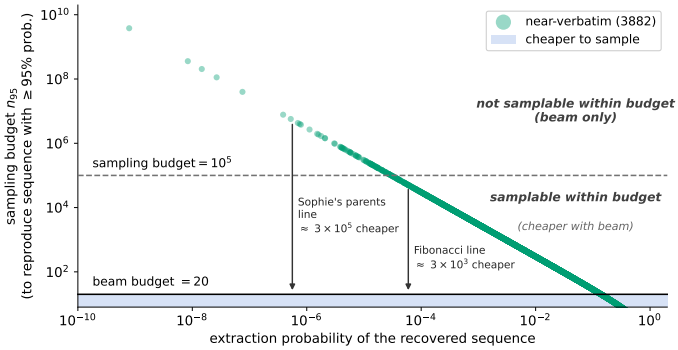


Figure 22: **Sampling cost vs. deterministic near-verbatim recovery.** Comparing the budget required to sample a memorized sequence from *The Da Vinci Code* with the cost of surfacing it using the beam-search procedure. The two annotated sequences show how much cheaper it is to recover them with this algorithm. The plotted points use the lower-bound near-verbatim ε-ball probability.

G ASSUMPTIONS USED THROUGHOUT THE PAPER

This appendix collects the assumptions behind our results in one place. Everything here is stated elsewhere in the paper, in the context of where it's used. We organize and repeat everything here for readability. For each assumption, we say what it requires for the conformal test in our population experiments (§3, §4) and what (if anything) it requires for the census (§5).

Matched controls.

- *Conformal test.* The conformal test assumes that the candidate's score and the control scores are exchangeable under the matched null (§3, §C.2). That is, a non-member candidate's score looks like one more draw from the control scores' distribution. This requires that the controls be selected by the same rule that would select the candidate if it were a non-member; that they share the candidate's source, format, length, and other score-relevant features; and that they actually be non-members (§C.3). Exchangeability doesn't require the controls to be independent of one another or of the candidate; our experiments draw the controls i.i.d., which implies exchangeability, and the Beta law and the intervals discussed below additionally require independence (§C.2). Matching is a best effort, and §C.3 discusses its practical limits; contamination of the controls (i.e., the control sample actually contains members) can only make the test more conservative (§D.2).
- *Census.* We run experiments on pairs of books. The member book comes from Books3, and is known to be in Llama 3.1's training data. The matched control is a post-cutoff book by the same author; matching on author is our best effort at exchangeability (§5). Contamination has to be handled differently here, because $\tau^{(\text{book})}$ is set via the single maximum control score. One highly memorized, contaminated passage in the control book (e.g., a very popular quote) can make $\tau^{(\text{book})}$ close to 1, so we decontaminate the control's high-scoring tail. We don't cull the in-training book, since we want to take a census that covers the entire book (§E.1).

A fixed procedure.

- *Conformal test.* The target FPR α and the complete extraction procedure (model θ, decoding policy ϕ, prompt-selection rule $f_{\mathbb{P}}$, score score, and the form of verify) are fixed before any control or candidate score is observed; the threshold τ_{α_N} is set from the controls (§3, §C.2).
- *Census.* The same requirement of a fixed procedure holds. In our experiments, we use two procedures: in one set, the sliding-window verbatim probabilistic extraction of Cooper et al. (2026a) under top-k decoding ($k = 40$); in another, the same sliding-window procedure, scored with the lower-bound algorithm for near-verbatim probabilistic extraction from Cooper et al. (2026b) (also with $k = 40$). Both are applied identically to the member book and its matched control (§5, §E, §F.3). However, the census differs from the conformal test with respect to what gets calibrated from the control and what is claimed. No population is sampled, so there's no α and no FPR to certify. Instead, the control book sets $\tau^{(\text{book})}$, the measured maximum on the decontaminated control scores. As a result, by construction, there are no false positives among the control windows. But this shouldn't be confused with a certified FPR of 0, since there is no notion of an FPR here (§E.1).

Ties and atoms.

- *Conformal test.* Prop. 1 doesn't require the score to be continuous. If the $N+1$ scores don't have ties, the marginal FPR equals α_N exactly; ties can only make the test conservative (§3, §C.2). Continuity matters for the Beta law of the population FPR α^* (Prop. 2), which also assumes that the scores are i.i.d. Our population experiments draw the controls i.i.d. (§C.2). Some of the decoding policies we use produce atoms: under top-k, a candidate that isn't reachable from the top-k tokens at every step has score exactly 0; greedy decoding results in a binary score (§D.4). Lemma 3 extends the Beta law to scores with atoms, including the atom at 0 that top-k decoding produces (§C.4.2). In our experiments, the nonzero control scores are all distinct (§C.4.4). Under greedy decoding, α^* isn't Beta-distributed and we don't draw the Beta band (§C.4.4).
- *Census.* Not applicable. $\tau^{(\text{book})}$ is a rounded-up maximum, not a certified level (§E.1), so there's no FPR for ties to affect (and, accordingly, no Beta law).

Finite controls.

- *Conformal test.* N controls can certify FPRs only in steps of $1/(N+1)$: $\alpha_N = K/(N+1)$ with $K = \lfloor \alpha(N+1) \rfloor$ (Eq. 5). The smallest nonzero certified FPR is $1/(N+1)$, reached at $K = 1$, where the threshold is the largest control score; setting $\alpha < 1/(N+1)$ gives $K = 0$ and the test can never reject (§3, §C.2).
- *Census.* Not applicable, since there's no notion of a certified FPR (see above).

Marginal guarantee.

- *Conformal test.* The guarantee of Prop. 1 is marginal: it averages over control draws (§C.4.3). For the specific control sample we draw, the calibrated threshold has a population FPR α^* that changes with the draw, and the marginal guarantee doesn't say how far that α^* sits from α_N (§C.4). Saying anything about it requires the i.i.d. assumption above, under which $\alpha^* \sim \mathrm{Beta}(K, N - K + 1)$ with mean α_N (Prop. 2); Cor. 5 turns this into a guarantee conditional on the drawn control sample. Our population figures have two sources of randomness: the member sample and the control draw. The Clopper–Pearson intervals in our figures cover the randomness in the member sample, holding the calibrated threshold fixed; they deliberately leave out the randomness in the control draw, since the plotted quantity is defined with the constant α_N, not the unknown α^* (§D.1). For the second source of randomness, the Beta law gives the distribution of α^* across control draws, which is fixed before any control is drawn and isn't a confidence interval (§C.1).
- *Census.* Not applicable, since there's no notion of a certified FPR (see above).

Many candidates, one control sample.

- *Conformal test.* The guarantee of Prop. 1 is per candidate: for each candidate, the test either claims memorization or makes no claim, and nothing in the guarantee requires the candidates to be independent of one another (§C.5). One control sample can therefore serve to make calibrated claims for many candidates, and each claim stands on its own. Reusing the sample does make the claims dependent on one another, since every candidate is tested against the same threshold τ_{α_N}. For a continuous score, the outcomes for two non-member candidates drawn independently of each other and of the controls are positively correlated, at exactly $1/(N+2)$ (Bates et al., 2021). The per-candidate guarantee doesn't by itself bound the familywise error rate or the false discovery rate over many candidates (§3); §C.5 discusses both, and we don't study either bound in this work.
- *Census.* Every window of the member book is compared against the same $\tau^{(\mathrm{book})}$, and the windows overlap heavily (§5). Coverage is the fraction of the book's characters spanned by windows above $\tau^{(\mathrm{book})}$ (Eq. 9).

Intervals.

- *Conformal test.* In our Wikipedia experiments, we draw candidates and controls uniformly at random within their date ranges, so each is an i.i.d. sample (§4). The member rate above a threshold is then a binomial proportion, and the intervals we plot on it are exact Clopper–Pearson intervals (§D.1). The calibrated rate subtracts the constant α_N, so its interval is the Clopper–Pearson interval translated down by α_N, with the same width. We instead use Newcombe's independent-proportions interval when the subtracted quantity is estimated from the controls (the greedy in-sample floor, or the non-member rate at a fixed budget); for the same candidates scored by two models, we use Newcombe's paired-proportions interval. All plotted intervals are 95% and pointwise: each is a statement about one plotted point, not about a curve as a whole.
- *Census.* Every window of each book is scored, so the coverage we report is the exact value for that book rather than an estimate, and none of our census results carries an interval (§E).

Budget views.

- *Conformal test.* The budget views convert a per-query generation probability into the number of attempts needed to generate the candidate at least once with probability c, which we set to 95%

throughout (Eq. 32; §F.1). For a fixed prompt, each attempt is an independent sample from the decoding policy, so the budget follows directly from $p = \mathsf{score}(\boldsymbol{s})$. The population budget n_{α_N} is computed from the calibrated threshold τ_{α_N}, so it inherits the assumptions above (§F.2).

- *Census.* The census budget views use $\tau^{(\text{book})}$ in the same way, calibrated separately for each extraction procedure, so the verbatim and near-verbatim results for the same book pair can use different thresholds (§F.3).

H Extended discussion of prior work

This appendix provides an extended discussion of relevant prior work. We first discuss in greater detail how existing extraction methods are instances of our definition of an extraction procedure (Def. B.2; §H.1). We then revisit two well-cited definitions of extraction (§H.2) and examine the validity of memorization claims in some recent work (§H.3) in light of matched comparisons. Finally, we describe similarities and differences between our conformal test for memorization and membership inference attacks (§H.4).

H.1 Prior work on extraction, in these terms

Existing extraction methods are instances of Def. B.2. For verbatim **discoverable extraction** (Lee et al., 2022; Carlini et al., 2023), θ is an open-weight non-instruction-tuned model, ϕ is greedy decoding, and $f_{\mathbb{P}}$ returns a singleton: $\boldsymbol{p}$ (Eq. 13) is the (typically 50-token) natural prefix of $\boldsymbol{s}$. The model is prompted with $\boldsymbol{p}$ and greedily generates an ℓ-token continuation $\boldsymbol{y}$. Since $q_{\theta,\mathsf{greedy}}(\cdot \mid \boldsymbol{p}) \in \{0, 1\}$, the score reduces to the binary match indicator $\mathsf{score}(\boldsymbol{s}) = \mathbf{1}[\boldsymbol{y} = \boldsymbol{s}]$ (Eq. 14), and verify deems $\boldsymbol{s}$ memorized when it equals 1. The relaxation of discoverable extraction, **probabilistic (discoverable) extraction** (Hayes et al., 2025b), follows the same setup, but makes ϕ stochastic (e.g., top-k) and replaces the binary score with $\mathsf{score}(\boldsymbol{s}) = q_{\theta,\phi}(\boldsymbol{s} \mid \boldsymbol{p})$ (Eq. 10), computed exactly using teacher forcing rather than estimated with sampling. Hayes et al. (2025b) leave verify fairly informal; as a sanity check, they run one experiment to show that the same model generates unseen test data at rates orders of magnitude lower than training data. By contrast, Cooper et al. (2026a) run a suite of experiments that score non-training data; these experiments determine a conservative global threshold $\tau_{\min}$, where verify evaluates $\mathbf{1}[q_{\theta,\phi}(\boldsymbol{s} \mid \boldsymbol{p}) \geq \tau_{\min}]$.

Both discoverable extraction and its probabilistic relaxation extend naturally to the near-verbatim setting, in which verify considers extraction successful when there is a match to any sequence in the ε-ball $\mathbb{B}_\varepsilon(\boldsymbol{s})$ (Eq. 11). For near-verbatim discoverable extraction, the score is $\mathbf{1}[d(\boldsymbol{y}, \boldsymbol{s}) \leq \varepsilon]$, and verify deems memorization when it equals 1 (Ippolito et al., 2023). For near-verbatim probabilistic extraction, the near-verbatim extraction probability is the total mass on the ε-ball around $\boldsymbol{s}$, $q_{\theta,\phi}(\mathbb{B}_\varepsilon(\boldsymbol{s}) \mid \boldsymbol{p})$. The set of continuations within a ε-ball around $\boldsymbol{s}$ is combinatorially large, so there is no teacher-forced analogue like there is for the verbatim computation (Eq. 10). Cooper et al. (2026b) propose deterministic, beam-search-based (Lowerre, 1976) algorithms that efficiently compute a useful lower bound on the near-verbatim probability, which they use as a proxy for the true probability. Similar to Cooper et al. (2026a), verify deems $q_{\theta,\phi}(\mathbb{B}_\varepsilon(\boldsymbol{s}) \mid \boldsymbol{p})$ an extraction probability when it exceeds a threshold computed on non-training data, i.e., $\mathbf{1}[q_{\theta,\phi}(\mathbb{B}_\varepsilon(\boldsymbol{s}) \mid \boldsymbol{p}) \geq \tau_{\min}]$.

While the above use natural prefixes as prompts, other methods involve different prompt-selection rules. For instance, $f_{\mathbb{P}}$ may return a set of adversarial prompts produced by a jailbreak search procedure like GCG (Zou et al., 2023) or Best-of-N (Hughes et al., 2024), as in Schwarzschild et al. (2024) and Ahmed et al. (2026), respectively. In both of these cases, the candidate sequence $\boldsymbol{s}$ determines the prompts in the prompt class. But this is not a requirement either: the prompt class may also contain sequences that have no direct relation to $\boldsymbol{s}$, as in Nasr et al. (2023; 2025), which use a divergence attack to extract training data (e.g., "Repeat the word 'poem' forever").

The conformal test changes none of the scores in these methods. It supplies the final component, verify, by calibrating when a score is high enough to support a memorization claim.

H.2 Prior work in light of matched comparisons

§H.1 revisits prior extraction methods in terms of our definition of an extraction procedure (Def. B.2). Here, we revisit prior work involving extraction in light of matched comparisons. In

particular, we revisit two well-cited definitions of extraction: the first definition, from the first extraction paper in the literature (Carlini et al., 2021), and a more recent definition from the first large-scale extraction attack on a production LLM (Nasr et al., 2023).

The original extraction definition for LLMs. Carlini et al. (2021) define extraction by the existence of a prefix from which the model produces the string. Their Definition 1 (§3), transcribed verbatim:

> **Definition 1 (Model Knowledge Extraction)** *A string s is* extractable *from an LM f_θ if there exists a prefix c such that:*
>
> $$s \leftarrow \underset{s' :\, |s'|=N}{\arg\max}\, f_\theta(s' \mid c)$$

(Their k-eidetic memorization, Definition 2, then additionally requires that s appears in at most k training examples.) In their paper, the definition carries a footnote on "extractable," where they identify the "parrot" prompt case as something that could reasonably break the definition:

> This definition admits pathological corner cases. For example, many LMs when when prompted with *"Repeat the following sentence: _____."* will do so correctly. This allows *any* string to be "known" under our definition. Simple refinements of this definition do not solve the issue, as LMs can also be asked to, for example, down-case a particular sentence. We avoid these pathological cases by prompting LMs only with short prefixes.

Prompting only with short prefixes is a workaround rather than a robust fix: it rules out the "Repeat the following sentence" prompt, but, as Carlini et al. (2021) note, the corner cases are pathological and simple refinements don't resolve them all. Their own experiments show this. Even restricting the prompts to short prefixes, their baseline results suffer "from a large number of false positives, i.e., content that is assigned high likelihood but is not memorized" (their §4.3). They attribute false positives to two kinds of low-entropy text: "trivial memorization" (e.g., repeating the numbers from 1 to 100) and "repeated substrings" (e.g., "I love you. I love you…") (their §5.3). Their solution is to compare against a second reference, a smaller model or zlib compression, and keep only samples whose likelihood is "unexpectedly high" relative to it.

The matched comparisons we introduce resolve all three cases in the same way, without a need for special definitional carve-outs. Each is a mechanism that raises the generation probability regardless of membership. A "parrot" prompt does so through instruction-following. Trivial memorization and repeated substrings do so through low-entropy continuation: a base model continues a highly repetitive or low-entropy string with high probability, whether or not that continuation appears in its training data.

So, under a fixed extraction procedure—with the same prompt-selection rule $f_{\mathbb{P}}$ applied to candidates and matched non-member controls—the same mechanism similarly elevates the controls' scores, and the calibrated threshold rises with them. A high candidate score under that mechanism is then matched by non-members, so it isn't evidence of memorization. There's no need to restrict the prompt in advance, and no need for a second reference to filter "unexpectedly high" likelihoods: the matched non-member null is the reference; we don't need special filters designed for each failure mode. (We've checked the low-entropy cases informally on Llama 3.1 70B base, with "I love you I love you I love you", a repeated common phrase, and "green pizza telescope green pizza telescope…", which is repeated nonsense. Both are produced with high probability regardless of training-data membership, so a matched control scores as high as the candidate. We haven't run this at scale.)

Extraction with an unrestricted prompt. Nasr et al. (2023) instead place no restriction on the prompts used to elicit extraction. They allow for any prompt an adversary can construct without direct access to the training data. Their Definition 1 (§3.1):

> **Definition 1 (Extractable memorization).** *Given a model with a generation routine* Gen*, an example $\boldsymbol{x}$ from the training set $\mathbb{X}$ is extractably memorized if an adversary (without access to $\mathbb{X}$) can construct a prompt $\boldsymbol{p}$ that makes the model produce $\boldsymbol{x}$ (i.e.,* $\mathsf{Gen}(\boldsymbol{p}) = \boldsymbol{x}$*).*

The generality of this definition is deliberate: a footnote to the definition says it's meant to be "a general definition that encompasses attacks in this work and our own," including prompts unrelated to the candidate, like their divergence attack (§H.1). As a result, the whole family of "parrot" prompts satisfies this definition, as well.

Matched comparisons can similarly fix the issues in this definition, without narrowing it. The definition can still allow any prompt the adversary constructs, so long as the same prompt-selection rule is also run on matched non-member controls. If that rule also makes matched non-members appear "extracted," then a hit on a training example isn't evidence of memorization.

H.3 Validity in prior work

As we show in this work, a memorization claim is valid when the candidate's score clears a threshold calibrated on matched non-members under the same extraction procedure (Box 1; §3; §C; §E); without running a baseline on non-members, it's impossible to know if a candidate's high score could be due to more general predictability.

Running matched comparisons like this is consistent with the field's preference for reporting memorization claims at low FPRs, even at the cost of false negatives (§H.4). This posture is implicit in the first extraction paper in the literature, which treated false positives as a serious problem its attack had to fix (Carlini et al., 2021), and was made explicit in later work on membership inference (Carlini et al., 2022).

Prior work has generally treated matched comparisons, when it runs them at all, as informal sanity checks (§1) (Lee et al., 2022; Carlini et al., 2023; Hayes et al., 2025b; Pappu et al., 2025; Schwarzschild et al., 2024; Cooper et al., 2026a;b). Below, we examine memorization claims in some recent work in light of matched comparisons.

Short candidates without matched comparisons. As our results show, a short verbatim match when generating training data shouldn't be treated as sufficient evidence for carrying a valid memorization claim on its own. At short candidate lengths, matched non-members can also be generated by capable models at very high rates, indicating many false positives (§4, §D.3). A lot of recent work nonetheless reports extraction using short matches, and does so without running a matched comparison (Chang et al., 2023; Karamolegkou et al., 2023; Wei et al., 2025; Liu et al., 2026). These claims rest on member generation rates, which can't separate memorization from general predictability on their own (§7). Liu et al. (2026) is a recent example. Its headline metric, book memorization coverage (bmc@5), counts every run of *five* or more consecutive words as evidence for successful extraction, where a "word" is an NLTK (Natural Language Toolkit) `wordpunct` token in their released code. Five NLTK words is far below the field's standard minimum of 50 tokens (approximately 37 readable words) (Carlini et al., 2023; Nasr et al., 2023; 2025; Hayes et al., 2025b).

Sanity checks. Some work does run negative controls, but not ones that match the claims being made. Schwarzschild et al. (2024) validate ACR with what they call "sanity checks that we hope any useful definition of memorization to pass": 100 random token strings of 3 to 17 tokens, and sentences from Associated Press articles published after the training cutoff, neither of which their procedure ever compresses. On that basis, they write that "while the GCG algorithm can elicit any output, we never observe compression of non-training data. This suggests that our method is robust against false positives." But these controls run only on Pythia models, which the paper discloses as a limitation, whereas the headline results run on Llama-2-7B-chat and Phi-1.5 with no control at all;[12] and neither control is matched to the famous quotes the claims are about. A matched comparison would run the same procedure, on the same model, on matched non-training text. We ran one check on Llama 2 13B and were successful in generating non-training data (§A.2).[13]

Prompt leakage. Another problem in some recent work is that the prompts used to elicit memorized training data can themselves supply words that get counted as a match in generated outputs. For

[12] While the models for their headline results don't have publicly disclosed training datasets, enough is publicly disclosed about their training (e.g., Llama 2 models on Books3 (Cooper et al., 2026a), the training cutoffs) to run on non-member sequences.

[13] As noted in §A.2, we identified other issues in the ACR implementation that required us to change the GCG success condition in order to reliably register success on training data, not just non-training data.

instance, nothing in GCG prevents the optimized prompt from leaking the target. When we ran ACR, we found that the optimized prompts often contained words from the target itself, on both training targets and on the post-cutoff quotes (§A.2). Leakage is also potentially responsible for some of the short-span results that go into computing $\mathrm{bmc@5}$ in Liu et al. (2026). Each prompt is a detailed plot summary of the excerpt under test; such prompts supply the passage's characters, events, their order, and distinctive words and short phrases. The authors note this, and add a trimming step that removes contiguous five-word overlaps between an output and its prompt. But it's possible that short five-word matches in outputs could be reassembled from words the prompt's summary supplied, even if the prompt doesn't contain those short matches contiguously. This is similar to what we observe for the non-training Jalen Brunson quote in our ACR control (§A.2). This type of leakage would survive the paper's trimming procedure. The paper doesn't run controls on non-training data to test for potential false positives (either with respect to counting very short spans or prompt leakage).

Validity issues in claims in the opposite direction. The above addresses work that risks overclaiming memorization, due to omitting matched comparisons to assess potential false positives. Other recent work runs the other direction; it claims that extraction is unreliable evidence of memorization, because models can verbatim complete text they weren't explicitly trained on (Liu et al., 2025; Morris et al., 2025). In their experiments, neither Liu et al. (2025) nor Morris et al. (2025) removes near-duplicates of training data from the sequences it treats as non-members; such near-duplicates are functionally members of the training data (Eq. 2, Lee et al. (2022)), so a verbatim completion (especially of famous passages, like quotes from the U.S. Constitution (Liu et al., 2025)) quite reasonably reflects memorization rather than being a counterexample to the validity of extraction (§C.3). And regardless of this issue, the conclusion is too strong: valid extraction claims are always relative to a specific extraction procedure and baseline, so a negative result under one procedure says nothing about calibrated claims under another (§7).

In each of these cases, the fix is the same: when running extraction experiments, one needs to run a matched comparison to make a valid claim (Box 1). That is what makes a memorization claim valid at the low FPR the field prioritizes (§H.4), and it's also what the conformal test and the matched census in this paper are constructed to do.

H.4 Membership inference

Our conformal test is a **membership inference** test (Shokri et al., 2017). Its null hypothesis is that the candidate $\boldsymbol{s}$ is a matched non-member, and rejecting the null says that $\boldsymbol{s}$ is a member that the model memorized (§3). Membership inference predates work on memorization and extraction in language models, and the strongest **membership inference attacks (MIAs)** rely on reference models (Shokri et al., 2017; Carlini et al., 2022). Reference-model MIAs ultimately ask the same overarching question as our test: the null hypothesis is that the model under evaluation was trained without sequence $\boldsymbol{s}$. The likelihood-ratio attack (LiRA) of Carlini et al. (2022) estimates the distribution of the model's score (e.g., likelihood) on $\boldsymbol{s}$ under two sets of reference models—trained with $\boldsymbol{s}$ and trained without it—by training many models on random samples of the data distribution and fitting a Gaussian to each set's scores; the MIA rejects the null when the likelihood ratio of the observed score under the two fitted distributions exceeds a calibrated threshold. (The offline variant of the attack uses only the reference models trained without $\boldsymbol{s}$, and rejects when the observed score falls in the upper tail of that set's fitted distribution.)

Data versus models. Our test and reference-model MIAs differ in what the null reference is composed of. The conformal test compares $\mathsf{score}(\boldsymbol{s})$ against the scores of matched non-member *data*, under *one fixed model*; reference-model MIAs compare the score of the model under evaluation on *a given sequence* $\boldsymbol{s}$ against the scores of reference *models* trained with and without it. Everything discussed in this subsection has to do with this difference in the null hypotheses that the two tests use to address the same question: whether the model memorized its training data. (Membership inference in general doesn't require that the memorized training data can be *generated* from the model; that requirement is specific to extraction (Carlini et al., 2021).)

Testing known members. In extraction experiments (including ours), the candidates are often known members. The membership half of a rejection of the null hypothesis is then trivial, but the memorization part of the claim remains. A score that clears the calibrated threshold is too high to be explained by general predictability, so the candidate *behaves like a member*: its score is one that

matched non-members produce only at the certified FPR (§C.5). Members whose scores fall below the threshold are of course still members; the test just makes no memorization claim about them.

Exchangeability over data versus over models. Each test rests on an exchangeability assumption, and what has to be exchangeable depends on how the null is set up. For the conformal test, the candidate $\boldsymbol{s}$ must be exchangeable with the controls: selected by the same rule and scored under the same fixed model, so that a non-member candidate's score would look like one more draw from the control scores (§3, §C.2). For a reference-model MIA, the model under evaluation must be exchangeable, under the null, with the reference models trained without $\boldsymbol{s}$. Those reference models estimate the null distribution only if the model under evaluation is one more draw from the same family of models; otherwise, the tail probability the offline test computes isn't the FPR for that model. (In the online variant, the reference models trained with $\boldsymbol{s}$ play the same role for the alternative; with the threshold set against null draws, they govern the attack's power rather than its FPR.) This exchangeability assumption is latent in the foundational work on MIAs (Carlini et al., 2022), but LiRA (and other reference-model MIAs like RMIA (Zarifzadeh et al., 2024)) depends on it regardless.

In both MIAs and our conformal test, exchangeability is a best effort in practice (§C.3). But where the effort is located differs. For the conformal test, we choose the selection rule and draw the controls ourselves, and the model is the same fixed object for candidate and controls. Distribution shift or contamination of the controls degrades the quality of the match (§C.3, §D.2). In reference-model MIAs, an attacker trains many models to estimate the distribution that the model under evaluation is assumed to be a draw from. In practice, training has various sources of non-determinism (Hayes et al., 2025a; Jebreel et al., 2026), and not all of it is random: for instance, hardware non-determinism (Pham et al., 2020; Zhuang et al., 2022) can compromise exchangeability.

Rank-based vs. parametric calibration. The exchangeability assumption also determines what kind of calibration the null supports. If the model under evaluation is exchangeable with N reference models trained without $\boldsymbol{s}$, then thresholding at an empirical quantile of their scores has a finite-sample false-positive guarantee, just as the conformal test does, with a floor of order $1/N$. Ye et al. (2022)'s Attack R follows this setup: its threshold for $\boldsymbol{s}$ is the α-percentile of $\boldsymbol{s}$'s loss over reference models trained on i.i.d. samples of the population data, which they describe as approximately guaranteeing FPR α. (Their construction is also where the exchangeability assumption comes closest to being stated among the papers discussed in this appendix: the reference models are explicitly i.i.d. draws, and what remains implicit is that the model under evaluation is another such draw.) Carlini et al. (2022) don't threshold at the empirical quantile; they instead fit a Gaussian to each set of reference models' scores and compute the test statistic from the fitted distributions rather than from the scores' ranks.[14] The Gaussian is a parametric assumption: with 256 reference models an empirical quantile can't resolve levels below $1/256 \approx 0.4\%$, but the fitted distribution has a tail at every level, so the statistic extrapolates past the data and the attack can be run at 0.1% and below. A false-positive level taken from the fitted tail is then only as good as the fit. In contrast, the conformal test uses the $1/(N+1)$ grid that N controls can certify, with marginal FPR at most α_N (§3), and assumes nothing about the shape of the control distribution; certifying a smaller floor requires more controls, not extrapolation (§4).

ROC curves and AUC. MIA results are commonly reported as ROC curves and summarized by AUC (Carlini et al., 2022). Our sweep (Fig. 4) isn't an ROC. Even when the score can reach an FPR of 1 (e.g., under $T = 1$ sampling, where every suffix has non-zero probability; §D.4), no threshold ever certifies an FPR of 0 (since N controls certify levels only down to $1/(N+1)$; see §3). And there's no true-positive rate (TPR) for the memorization part of the claim, since which members are memorized isn't known (Def. 4.1). An ROC curve needs both at every operating point: an FPR that runs from 0 to 1, and a TPR to plot against. (AUC then integrates the curve over that whole span.)

The field's posture for making memorization claims at low FPR. In this paper, we privilege a low FPR, even at the cost of false negatives (under-counting true instances of memorization). This posture is inherited from foundational work on memorization. Carlini et al. (2022) evaluate LiRA attacks by their true-positive rate at FPRs of 0.1% and below. Carlini et al. (2022) also make this point about extraction directly:

[14] "By design, the exact test of Ye et al. can at best target an FPR of $1/N$ with N shadow models. It is thus inapplicable in the setting we consider here (256 shadow models, and a target FPR of 0.1%)" (internal citations omitted) (Carlini et al., 2022).

> to extract training data it is far more important for attacks to have a low false positive rate than high average success, as false positives are fare [sic] more costly than false negatives (internal citations omitted).

It's from this lineage that we draw our paper's name (Carlini et al., 2022), and this conservative posture runs through both of our experimental settings. For the population experiments, the conformal test is a hypothesis test whose calibrated threshold limits false positives to a certified level, and the low-FPR end of the sweep is where one would operate in practice (§3, §4). The census has no FPR to certify (§5), but it inherits the same conservatism: $\tau^{(\text{book})}$ is set at the largest score any control window attains. This is also why we wrote this paper. Recent work in the extraction literature studies settings that risk false positives, but often doesn't run the appropriate control baselines. Without those baselines, there's no way to tell how much false positives affect the validity of memorization claims; such work may have high false positives, which is contrary to what the field accepts when making memorization claims (§H.3).