Watermarking for Proprietary Dataset Protection

John Kirchenbauer [1] **Brian R. Bartoldson** [2] **Bhavya Kailkhura** [2] **Tom Goldstein** [1]

Abstract

A growing body of literature suggests that training data membership inference problems are fundamentally hard tasks in modern language modeling settings. We argue that output watermarking techniques are the right gadget to make training membership tests for generative models more tractable, based on prior results showing that language models exhibit residual watermark "radioactivity" under partially watermarked training datasets. We pit a watermark-based dataset inference approach head-to-head against traditional loss-based membership inference methods and show that watermarking can achieve comparable membership detection performance when subset exposure is high enough, under an alternate set of assumptions.

1. Introduction

Modern language models perform complex knowledge work of growing economic value, but the regulatory frameworks governing fair use of the web-scraped data they train on remain underdeveloped. Recent litigation suggests content owners like news websites and independent authors may be entitled to compensation for inclusion of their datasets in large-scale AI model training. Answering such data-use questions in high-stakes settings requires a concrete definition of what it means to test whether some data was included in a model's training dataset.

While the fully general training data attribution problem asks how a model's test-time behaviors are caused by specific training instances, the question at hand in contemporary fair-use deliberations is actually just *membership*. Membership inference attacks (MIAs) ask whether a specific sample was in a model's training dataset; dataset inference attacks (DIAs) generalize this to whole collections. As the more relevant setting for IP and generative-model training disputes, in this work we study the DIA setup but refer to both problem types collectively as membership problems.

[1]University of Maryland [2]Lawrence Livermore National Labs. Correspondence to: John Kirchenbauer <jkirchen@umd.edu>.

Presented at the ICML 2026 Workshop on Trustworthy AI4GOOD, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

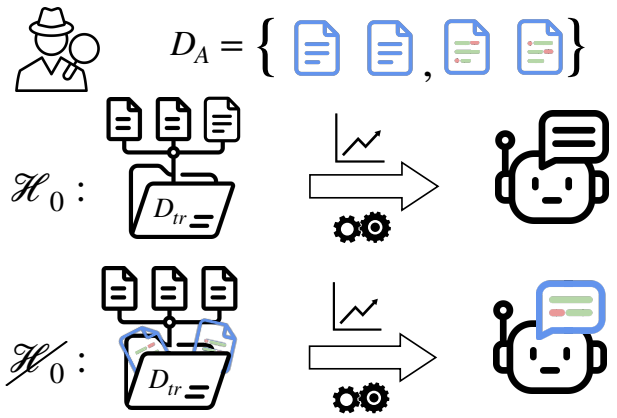

Figure 1. To protect a proprietary dataset from unauthorized use in training, the dataset owner (attacker) paraphrases their documents with a secret watermark key. To perform dataset inference, the attacker tests the suspect model's predictions for evidence of the watermark key. The watermark detection test is used to conclude whether their protected data was included in the training dataset.

What makes training dataset membership tests challenging in modern settings? Performing membership tests on modern generative models is harder than it was in discriminative settings for classifiers which mapped inputs to as few as 10 or 100 classes (i.e. the setting of Shokri et al. (2017)). The variable-length output space matches the input cardinality, making analysis complicated, and the billion-parameter sizes of modern models make classical attribution tools like influence functions difficult to apply (Koh & Liang, 2017; Ilyas et al., 2022; Park et al., 2023). Even the definition of "a sample" is arbitrary at pretraining scale (documents, sentences, words), and individual samples overlap heavily in terms of shared n-grams, which destabilizes existing membership tests (Duan et al., 2024). Modern generative models also memorize *and* generalize along both semantic and stylistic lines, blurring the line between membership and generation attributability: sample membership is often neither necessary nor sufficient for a target output to be produced (Liu et al., 2025), and memorization rates vary widely (Cooper et al., 2025).

Proactive Interventions for Reliable Dataset Inference. Our work attempts to circumvent some of the difficulties

described above by proactively but sparingly intervening during the construction of training datasets to make membership tests easier. The essential operation will be to precisely *mark* the training dataset of a model in a way that does not significantly impact performance. This intervention will give the marker (data owner) a significant informational advantage about how the marked elements of the training dataset are expected to influence the model's test-time predictions, and most importantly, how to probe for these effects.

The main technical gadget we will use is language model output watermarking of the kind proposed and studied in Kirchenbauer et al. (2023). This watermarked decoding algorithm will be layered on top of a "paraphraser" language model to create versions of candidate samples with the same style and semantics, but that produce significant p-values under a watermark detection test. Recent work has shown that if a target model is trained on watermarked samples, it will absorb the watermark key's signature and emit a shadow of the same signature at test time (Sander et al., 2024). Then, when the attacker probes a suspect model using the watermark detection test, sample membership in this model's training dataset is inferrable based on the p-value observed.

Relationship to Prior Work. We build our study directly on two prior works. The folding construction used as our finetuning backbone is adapted from Hayes et al. (2025), whose paired-data MIA evaluation we extend to full-subset membership (DIA). The proactive watermarking DI approach builds on Sander et al. (2024; 2025), who show that a target model trained on watermarked samples measurably reproduces the watermark's key-specific signature. We adopt their sensitive "reading-mode" detector that tests next-token predictions (argmaxes) conditioned on watermarked prefixes rather than fully rolled-out completions. We vary per-key support fractions and effective epochs on the marked subsets, replace the detection test's parametric tail with an empirical-null randomization test (Section 2.2), and benchmark against loss-based and reference-model baselines—raw loss (Yeom et al., 2018), argmax match, min-$k\%$ (Shi et al., 2024), zlib (Carlini et al., 2021), and rMIA-simple, rMIA, and LiRA (Carlini et al., 2022)—on the same paired trials (Section C.6). **Our goal is to unify these settings and methods to calibrate claims about whether watermark-based dataset membership testing is ready for use in practice.**

Contributions:

- We formalize the general threat model for watermark-based proactive dataset inference as a data-owner-centric security game, and contextualize prior work as studying a specific instance of this problem (Section 2).
- We propose a randomization-based variant of the watermark detection test that ensures p-value validity under interactions between pretrained checkpoints, intervention samples, and specific watermark keys (Section 2.2).
- We implement a unified experimental setting for evaluating traditional loss-based dataset inference scores and watermark-based approaches on the same paired trials, and compare the performance of each technique under its respective threat-model assumptions (Section 3.1).
- We ablate under-explored dimensions from previous studies: per-key support vs. repetition at fixed training budget, training from scratch vs. continued pretraining, and insertion schedule shape for the marked subset (Sections 3.1 and 3.2).

2. Methodology

Definition 2.1 (*Threat Model*: Proactive Dataset Inference for Data Owners.). Alice suspects that a model owner Bob will train a language model f_{tgt} on $D_{tr} := D_A \cup D_{web}$, where D_A is a portion Alice owns or controls and the rest is drawn from sources Alice does not control. Alice wants a score $d : f_{tgt}, \mathcal{D}_A \to (0, 1)$ that predicts whether Bob included any of D_A in D_{tr}. Alice is allowed to modify D_A using a *marking* transformation $T_m : \mathcal{X} \to \mathcal{X}_m$ before Bob accesses it. Attack success is evaluated against the ground-truth membership label $m \in \{0, 1\}$ indicating D_A's inclusion in D_{tr}. In the strictest setting, Alice has neither the compute to train shadow models for f_{tgt} nor a hidden shadow copy of similar data D'_A to calibrate the detector against.

Definition 2.2 (*Gadget*: Watermark-based Dataset Protection.). Alice instantiates T_m using a generative model $f_{wm} : \mathcal{X}, k \to \mathcal{X}_{wm}$ running an output watermarking scheme keyed by secret key k, whose detector $d(f_{tgt}, k, D_A)$ admits a test statistic $z \in \mathbb{R}$ and associated p-value. She marks all or part of D_A via f_{wm} relying on the assumption that Bob's f_{tgt} trained on samples $x \in \mathcal{X}_{wm}$ will produce new generations that yield significant detection scores under k. A decision function $g : \mathbb{R} \to (0, 1)$ on z then implements the membership score $d = g(z)$, with the p-value itself being the obvious choice. The null hypothesis is defined as $\mathcal{H}_0$: Bob's model f_{tgt} was trained such that $D_A \cap D_{tr} = \emptyset$, or more precisely, such that none of the samples in D_{tr} are marked with k. Assuming the canonical choice of $g(\cdot)$, if a p-value below α is observed, Alice rejects the null and concludes that Bob likely trained f_{tgt} on her dataset.

2.1. Target Model Assumptions

We assume that the target model's tokenizer is known and matches the paraphraser's. Prior work already shows that this assumption can be relaxed and watermark detectabil-

ity still preserved through common-token filtering (Sander et al., 2024; 2025; Xu et al., 2026), so we omit experiments along this axis here. We also assume that the attacker can query the target model for next-token-prediction probabilities conditioned on a prefix—the "reading mode" detection procedure of Sander et al. (2024). This is a costly assumption in practice if one is making calls to a production API for every token. However, the increased sensitivity of the data-forced reading-mode test is necessary at the support fractions we consider: naively rolled-out completions from a target model that has only been mildly exposed to the watermark signature do not yield low enough p-values for reliable detection. Improvements in detection that would let the attacker relax the reading-mode assumption are an obvious axis of future work but are out of scope here.

2.2. Ensuring P-value Validity

Prior work on watermark-based DIAs reports significant p-values under intervention and near-null values otherwise (Sander et al., 2025). However, our preliminary reproduction experiments suggested that interactions between the model checkpoint, the particular samples in the intervention dataset, and the specific watermark key used can invalidate the independence assumptions on which standard parametric tail bounds depend, even under careful de-duplication and especially when the reading-mode detector is in use. Therefore, we propose a standard randomization-based (permutation) hypothesis test variant that establishes a valid p-value based on an empirical null hypothesis under exchangeability assumptions. This method allows us to configure the resolution of the exact p-value and adds only a small computational overhead to the attacker's testing protocol.

Definition 2.3 (Watermark Key Randomization Test for Exact P-Values). Assume Alice's watermark secret k_i is sampled i.i.d. from a distribution of possible keys suitable for the PRF used by the watermark scheme $k_i \sim \mathcal{K}$. Under the null hypothesis, f_{tgt} has not been trained on data marked with k_i nor any other key k_j in $\mathcal{K}$[1]. Thus all keys $k_j \neq k_i$ are exchangeable w.r.t. the test statistic produced by the detector $d(f_{tgt}, k, D_A)$. To compute the exact p-value using the randomization test, Alice samples $M-1$ distinct additional keys, and then produces $z_j = d(f_{tgt}, k_j, D_A), \forall j \in M$ including her key k_i. The M scores are sorted in descending order to determine each key's resulting rank r_j (highest z yields smallest r). The exact p-value of the test is r_i/M, the rank of Alice's actual key amongst M. Under the null, due to exchangeability, the probability of observing her key at rank r_i or earlier in the list is exactly $P[r \leq r_i | \mathcal{H}_0] = r_i/M$, making this a valid p-value under no additional assumptions.

Test-time cost is controlled using the vectorized PRF implementation in `textseal` (Sander et al., 2025; Fernandez et al., 2025) to evaluate many watermark secrets in parallel for any given piece of text. We verify the validity of the resulting empirical null in practice by confirming p-value uniformity using a KS test (see Figures 4, 14 and 22). For p-values below the $1/M$ resolution of the empirical sample, we report an extrapolated empirical-Gaussian fit to the same null distribution where useful.

[1]This is a strong, but plausible assumption if a sufficiently large, private keyspace is used.

3. Experiments

Our experiments are built around a controlled data-folding design that allows us to mimic prior studies on membership and dataset inference in language models while simultaneously benchmarking the proactive watermarking approach. In a smaller finetuning regime, we systematically vary the level at which the model is exposed to the intervention data by modulating subset size and repetition during training, before moving to a larger 10B-token training regime where we ablate the insertion schedule of the marked subset and the choice of initialization (continued pretraining versus from-scratch).

Setup. In our experiments, we strive for depth rather than breadth. Thus, we use a single language model `Qwen3/Qwen3-1.7B` (Yang et al., 2025) in all experiments, either as a pretrained set of weights or as the architectural specification for from-scratch experiments. Similarly, we adopt a single dataset of $N = 1500$ natural-looking but semantically isolated documents designed to be used in controlled experiments on memorization: FictionalQA (Kirchenbauer et al., 2025). This dataset includes webtext-like documents in generic styles about totally fictional entities and **events**—collections of related documents about the same fictional scenario (an explanatory blog post, a news article about the event, etc.). Our *event-split* fold construction (Section 3.1) uses the event groupings present in the dataset: fold boundaries respect events so that documents about the same fictional event stay together inside a single fold. We mix our controlled intervention data into a base random subset of `allenai/dolma3_mix-150B-1025` (Olmo et al., 2025) with both sources shuffled and packed into length-4096 chunks.

Dataset Membership Testing. During each dataset inference simulation trial, for the watermarking methods, for positive cases we report the pooled, deduplicated reading-mode test statistic and corresponding p-value measured on each of the watermarked folds for each of the models that trained on it. For the negative cases, we compute the same score using all the same watermarked folds as probe data, but on the corresponding clean (twin) model trained on the unmarked version of the data. For the non-watermarking methods, the original clean samples from the fold are fed through the

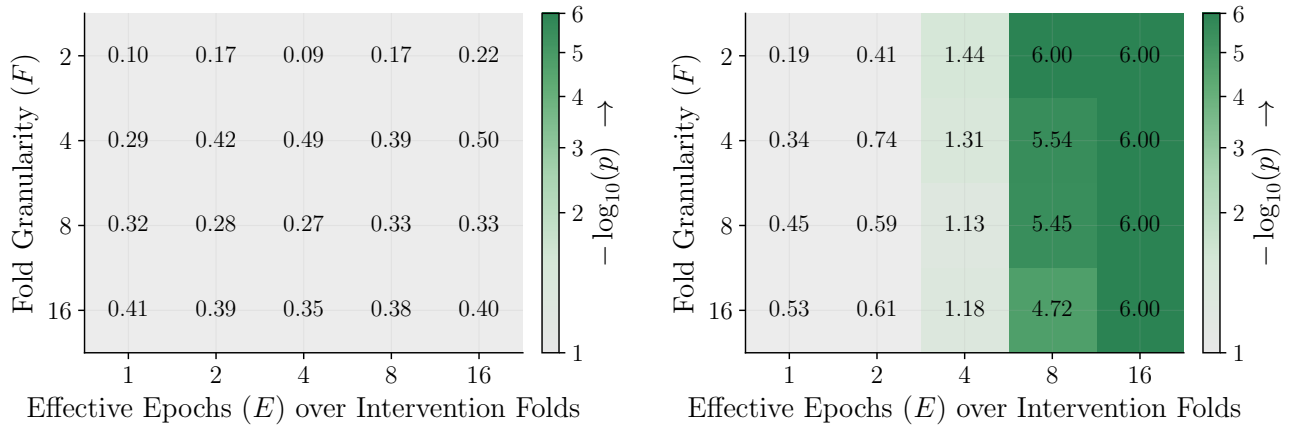

Figure 2. Finetuning event-split matched clean-twin false-probe null **(left)** and keyed signal **(right)** on the aligned unpacked detection surface, scored as $-\log_{10} p$ under an empirical-exact reference.

corresponding clean models to compute losses, and then each technique's corresponding algorithm is used to produce the row-level MIA or whole-model DIA score, potentially using the rest of the clean models as reference models. Throughout, we present results in terms of $-\log_{10} p$ for keyed signal strength, reporting the exact empirical-null reference distribution (Section 2.2) and/or an extrapolated empirical-Gaussian p-value where useful.

We score the watermark detection statistic under two surface variants. The **aligned** surface scores each original document in isolation and is the most realistic setting; Alice has no knowledge of exactly how her documents will be truncated or concatenated together. The **packed** surface scores the 4096-token chunks of multiple fictional documents packed exactly as they were at training time, and serves as an oracle baseline that quantifies how much keyed signal could be lost to train- vs. inference-time mismatches. We score the same readout under both an empirical-Gaussian and an exact empirical-null reference distribution (Section 2.2), and report headline results in $-\log_{10} p$ throughout (higher means a more significant signal).

Table 1. Idealized per-key exposure E/F for the finetuning grid, expressed as a multiple of the per-key support that a single epoch over a single fold would produce. Distinct watermarked tokens scale as $1/F$; epoch repetition E multiplies how many times those tokens are seen during training. The event-split and SKS regimes share this idealized table. Realized exposure based on the actual sampling rates during the training runs is reported in Sections C.4 and D.3.

	$E = 1$	$E = 2$	$E = 4$	$E = 8$	$E = 16$
$F = 2$	0.5	1.0	2.0	4.0	8.0
$F = 4$	0.25	0.5	1.0	2.0	4.0
$F = 8$	0.125	0.25	0.5	1.0	2.0
$F = 16$	0.0625	0.125	0.25	0.5	1.0

3.1. Finetuning: Varying Per-Key Exposure

In our finetuning-scale event-split design, multiple keyed folds of size N/F are mixed into each model's training data with the total watermarked share of the training tokens held fixed across F. As F grows, each individual key's per-fold footprint shrinks while the number of distinct keys observed by the model grows. We populate every cell of the $F \times E$ grid with enough watermarked-positive and matched clean-false-probe trials to make whole-model dataset inference comparisons stable. The per-cell training token budget is held fixed at 131M tokens. The interventional design is summarized in Table 1 (idealized per-key exposure E/F, shared with the SKS ablation); the matched realized normalized exposure $\hat{E}/F$ and the underlying realized $\hat{E}$ values live in the appendix (Tables 3 and 4). In absolute terms, the watermarked share of each model's training tokens swings from a low of about 71k tokens ($\approx$0.05% of the run) at (F=16, E=1) up to about 8.3M tokens ($\approx$6.3%) at (F=2, E=16), and per-cell whole-model DIA trial counts scale with F, from 2/2 positive/negative trials per cell at F=2 to 96/96 at F=16 (Tables 5 and 6).

Increased signal as a function of exposure (E/F). Figure 2 demonstrates the expected trends: increasing E produces strongly significant keyed signal whenever the underlying support is sufficient, while increasing F at fixed E weakens the signal as each individual key occupies less of the corpus (Figure 11 makes this more obvious). The DIA AUC view (Figure 3) tracks the keyed-signal ordering and saturates at 1.0 in the high-exposure corner, while the low-exposure corner sits near chance. The matching whole-model DIA comparison against loss-based and reference-model baselines on the same paired trials is reported in Table 2; the corresponding row-level MIA, matched false-probe-null, cross-key sham-null, and packed-surface coun-

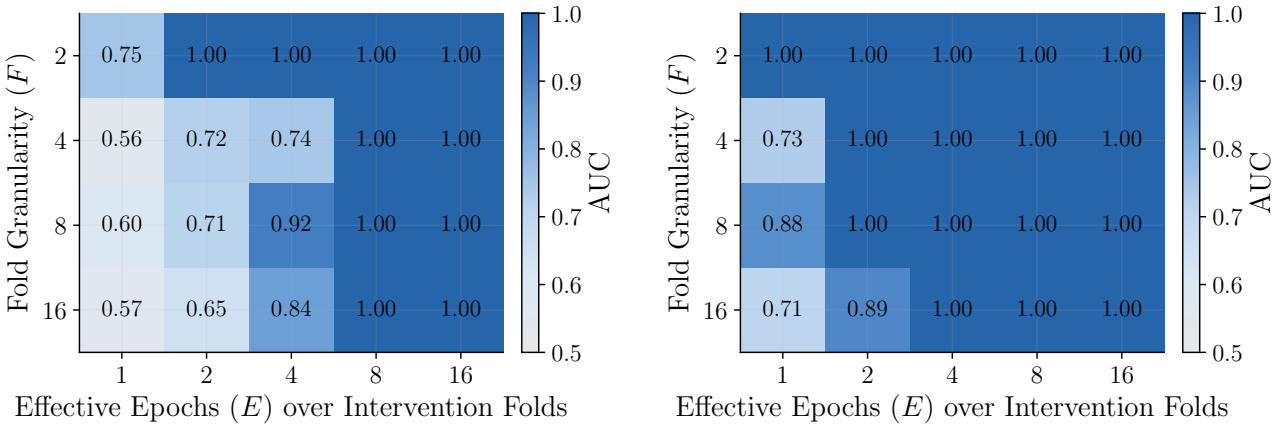

Figure 3. Finetuning event-split watermark whole-model DIA AUC across the $F \times E$ grid, on the aligned **(left)** and packed **(right)** detection surfaces, both scored against an empirical-exact null. The aligned chart on the left is based on the same probe surface as the right panel of Figure 2, paired here with the packed-surface companion. On the aligned surface the AUC saturates in the high-exposure corner; on the packed surface the more permissive oracle recovers several lower-exposure cells.

terparts are deferred to the appendix in Section C. The single-key support (SKS) ablation, which holds each model to one watermarked fold so per-model fictional support drops as $1/F$, is reported in Section D as a structurally simpler reference point.

Event-Split Null-Validity Figure 4 confirms that, once pooled across many distinct positive keys, the empirical exact null is close to uniform at the 1M-null scale and respects standard tail-rate thresholds, supporting the use of the empirical-null permutation test as the headline reference distribution for the keyed readout. Per-key idiosyncrasy is a separate concern visible in the right-hand trace of the extended chart Figure 14: at very small fold counts a single key can run systematically warm even when the pooled null is well-calibrated, and the inherited $F = 2$ scaffold reused in pretraining (Section 3.2) carries one such warm key.

Loss-Based and Reference-Model Baselines Table 2 reports the fold-level whole-model DIA comparisons for the event-split finetuning grid (for the decision problem "was model i trained on fold i?"), with the watermark method reported alongside the loss-based and reference-model baselines on the same paired trials. The watermark whole-model DIA AUC values reproduce those visualized in Figure 3. Row-level MIA performance for the loss-based methods is reported in Table 7. Using additional access to calibration samples ("Loss-based" ref-free) and/or the rest of the models in the pool trained and not trained on each fold as a reference pool ("Ref-model"), the baselines perform nearly perfectly with the exception of the very low-support $F8$ and $F16$ single-epoch cells. While valid to run above $F = 2$, LiRA still suffers in the $F4$ cells when the ref-model pool is too small for accurate in/out ratio estimation (see Table 6).

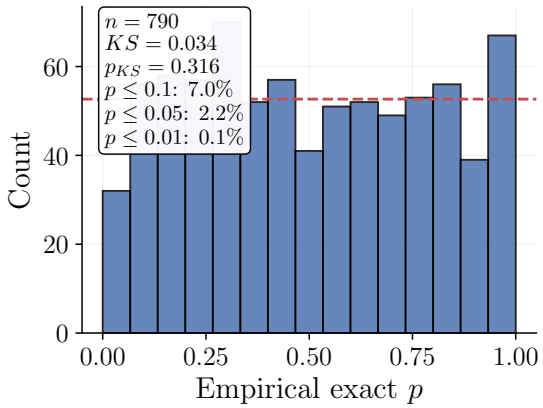

Figure 4. Event-grid null-validity panel. Histogram of pooled empirical exact p-values from the packed matched-clean-negative whole-model readings across the (F, E) event grid. The annotated statistic and p-value are from a one-sample Kolmogorov-Smirnov test of these pooled p-values against $\mathrm{Uniform}(0, 1)$; the dashed horizontal line is the expected per-bin count under a uniform histogram with the plotted binning.

We also render a compact summary table of access assumptions for the various approaches in the appendix (Table 8).

3.2. Pretraining: Sensitivity at Scale

To address whether the keyed readout remains detectable under much heavier dilution, we conduct a second, narrower batch of experiments at a 10B-token total budget. We reuse a similar $F = 2$ fold/key scaffold from finetuning (two groups of randomly sampled fictional documents of size $N/2$, simpler than the event-split construction) and sweep ten different insertion schedules for these two keyed folds.

The two initialization regimes are continued pretraining

Table 2. Event-split finetuning: fold-level whole-model DIA AUC comparison. Entries marked N/A indicate statistics that are not estimable in the available cell geometry; for example, LiRA in F=2 cells lacks sufficient in-reference models.

	Watermark DIA		Loss-based DIA			Ref-model DIA		
Cell	Aligned	Packed	Raw-loss	Argmax	min-k_{10}	rMIA-simple	rMIA	LiRA
(F2,E1)	0.7500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	N/A
(F2,E2)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	N/A
(F2,E4)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	N/A
(F2,E8)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	N/A
(F2,E16)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	N/A
(F4,E1)	0.5556	0.7292	1.0000	1.0000	1.0000	1.0000	1.0000	0.1181
(F4,E2)	0.7222	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.4861
(F4,E4)	0.7431	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.4514
(F4,E8)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.4514
(F4,E16)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8056
(F8,E1)	0.5968	0.8815	1.0000	0.9987	1.0000	1.0000	1.0000	1.0000
(F8,E2)	0.7118	0.9974	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(F8,E4)	0.9162	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(F8,E8)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(F8,E16)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(F16,E1)	0.5675	0.7062	0.9787	0.9702	0.9887	1.0000	1.0000	1.0000
(F16,E2)	0.6477	0.8949	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(F16,E4)	0.8439	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(F16,E8)	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(F16,E16)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

(CPT) from the Qwen3-1.7B checkpoint and from-scratch (random init) at the same total token budget. The four single-burst schedules $(S1, E1)$, $(S2, E1)$, $(S3, E1)$, and $(S4, E1)$ insert the watermarked fold once at one of four step locations within the run (target $E = 1$). The remaining six schedules pair a uniform-spacing variant (denoted U), which uses a constant sampling rate targeting $\sim E$ epochs by end of run, with a periodic-cluster variant (denoted P) that schedules E dumps of the entire fold at evenly spaced steps, very similar to Sander et al. (2025)'s setup. We cross these with $E \in \{4, 8, 16\}$. The idealized exposure profile of these schedules at $F = 2$ is summarized in Table 14; the matched realized normalized exposure tables and per-schedule realized $\hat{E}$ values for both initialization regimes also live in the appendix (Tables 17 to 20).

Main body figures use the exact empirical p-value for both CPT and from-scratch training. Each schedule contributes 2 watermarked-positive and 2 matched clean-negative whole-model trials per init, and (mirroring Sander et al. (2025)) the watermarked-token share of each 10.49B-token training run is small in absolute terms—about $0.5M$ tokens ($\approx 0.005\%$) for the four single-burst E=1 schedules, rising to about $8M$ tokens ($\approx 0.077\%$) at the E=16 schedules (Tables 21 to 24).

Schedule and Initialization Effects. Across both inits, schedule shape matters: the four single-burst E=1 schedules are not distinguishably higher than their null counterparts in Figure 5, while the multi-insertion U and P schedules become clearly separable from null as $\hat{E}$ grows. Under aligned testing the P and U schedules are similar, but the more sensitive packed-surface readings (Figures 24 and 25) show that the P setting produces a stronger signal for CPT. From-scratch recovers substantially stronger keyed signal than CPT at high $\hat{E}$, while at low $\hat{E}$ the two inits are comparable, though Figure 6 shows slightly more separability at $\hat{E} = 4$ for from-scratch than for CPT. Figures 29 and 30 show that the U sampling strategy produces more signal in the from-scratch setting, the opposite of the CPT result.

Fold/Key Asymmetry. The pretraining experiments reuse the same two folds and keys that the finetuning SKS $F = 2$ row was conducted with. At the finetuning scale, one of those two keys already produces systematically larger detection statistics than the other under matched-clean conditions, even when no watermark training signal is in play (right side of Figures 14 and 22). This handedness is visible again in Figure 5, where Key 0 consistently produces higher signal than the other under matched conditions (more obvious in Figure 24). This suggests that per-key interactions with the data and/or the pretrained checkpoint can produce scenarios where positive model signal is spuriously low or high, or false-positive rates could exceed what the nominal p-value threshold would indicate. It is possible that prior studies filtered for well-behaved watermark keys, and our systematic evaluation just reports this issue more transparently.

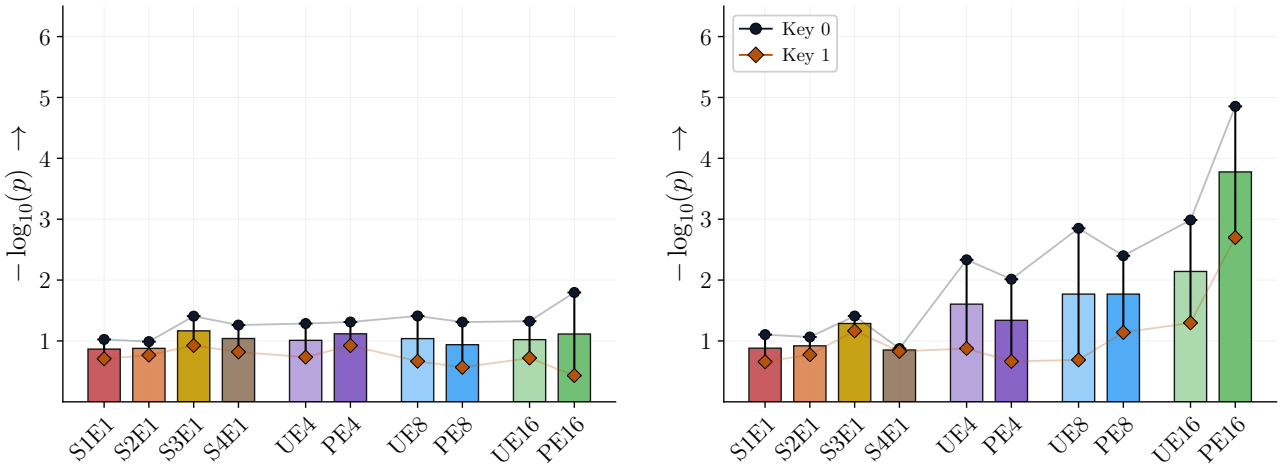

Figure 5. Pretraining signal across the ten-schedule sweep on the aligned unpacked detection surface, scored as $-\log_{10} p$ under the exact empirical-null reference, for the CPT initialization. **(Left)** is the clean-twin false-probe-null models and **(right)** is the keyed model trained on the watermarked data. Schedule clearly modulates the keyed readout: at $E1$, none of the keyed signals are separably significant, but at larger E a gap appears. One of the two inherited keys runs visibly warmer than the other in both panels. Matched packed-surface variants, extrapolated Gaussian p-value versions, and watermark whole-model DIA bar charts are all reported in the appendix (Section E.1)

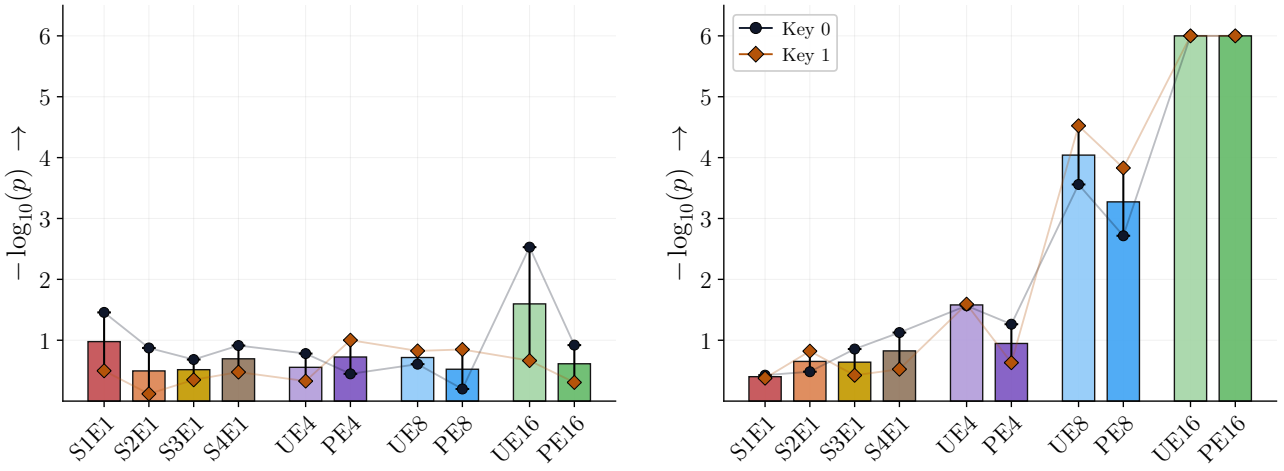

Figure 6. Pretraining signal across the ten-schedule sweep on the aligned unpacked detection surface, scored as $-\log_{10} p$ under the exact empirical-null reference, for the from-scratch initialization. **(Left)** is the clean-twin false-probe-null models and **(right)** is the keyed model trained on the watermarked data. The schedule strengthens the keyed readout much faster as a function of E in this setting versus CPT: at $E1$ there is a slight trend as a function of insertion point (Figure 29 makes this more obvious) at $E8$ a significant gap appears and $E16$ maxes out the exact resolution. Here, there is little if any systematic bias between the watermark keys. Matched packed-surface variants, extrapolated Gaussian p-value versions, and watermark whole-model DIA bar charts are all reported in the appendix (Section E.2)

Comparison to loss-based baselines. A complete loss-based and reference-model whole-model DIA head-to-head comparison for both pretraining inits is reported in the appendix (Section E and Tables 26 and 28). As in the fine-tuning regime, the loss-based approaches perform nearly perfectly across all pretraining schedules including the low-exposure $E1$ insertions. As the differences between Figure 6 and Figure 5 would suggest, the watermark-based approach is only able to separate the positives and negatives clearly when the models are trained from scratch at the higher-exposure $E8$ and $E16$ settings.

4. Discussion

As stated in the introduction, our setup unifies and builds upon two prior studies: Sander et al. (2025) and Hayes et al. (2025), rather than proposing a wholly novel methodology. Therefore, it is important to identify where our results co-here with and diverge from those studies. After presenting our analysis, we discuss the feasibility of Bob, the model

trainer, "defending" against this dataset protection strategy.

Watermark Detectability Trends. The trends we observe in both Section 3.1 and Section 3.2 on sensitivity growing with exposure to the watermarked subset match expectations from Sander et al. (2025). Even though that work does not directly ablate the size of the watermarked subset as we do at finetuning scales, we expect that, were resources sufficient to test it, the drop in significance as a function of increasing F that we observe in Section 3.1 would also appear at the pretraining scale.

Performance of Loss-based Methods. While Hayes et al. (2025) did not run as complete a controlled exposure grid as our (F, E) sweep in Section 3.1, the loss-based methods' row-level MIA success across the exposure grid varies in the very same way as the watermark signal strength (Section C.6). However, they almost perfectly solve the DI problem at all exposure levels anyway (Table 2). As noted in Hayes et al. (2025), LiRA struggles at low ref-model counts.

Sensitivity at Scale and FPR Control Issues. We observe relatively similar $-\log_{10} p$ values for $PE4$, $PE8$, and $PE16$ for the models trained on the watermarked data in Section 3.2 to analogous values reported in Sander et al. (2025) (our fold size vs. training budget are approximately matched here). However, this is only true when the model is trained from scratch, something Sander et al. (2025) did not ablate, and significance swings sharply depending on train- vs. detect-probe style (aligned/packed). The finetuning and CPT settings also reveal that a spuriously hot watermark key might confound the results or cause false positives (see Sections C.7 and D.6). The point estimates reported in Sander et al. (2025) make it hard to determine whether this was an issue in their study. However, they do report a $-\log_{10} p$ range of 0.3 to 0.9 in Table 1 at zero contaminations which is broadly in line with our Figure 6 null side readings, but crucially this is in the from-scratch setting where there is no preexisting pretrained checkpoint for the specific watermark key to potentially interact with. Our results show that even under a well-calibrated exact p-value test, spurious key-checkpoint interactions still matter for any single key in any single deployment attempt. Therefore, Alice may need to filter for "quiet" keys that behave well under the null hypothesis for representative data and models.

Can the Model Trainer Mount a Defense? The marking transformation $T_m : \mathcal{X} \to \mathcal{X}_m$ is intentionally notated to bring attention to the fact that $\mathcal{X}$ and $\mathcal{X}_m$ are distinguishable, *but only to Alice*. While a data owner does not necessarily need to mark their data in a way that satisfies a formal third-party indistinguishability condition (a la Christ et al. (2024)) for the attack to be successful, in practice, a marking scheme where the model trainer is unable to reliably (or *scalably*) determine whether or not a sample is in $\mathcal{X}_m$ is likely to be much more effective and tamper-proof. If Bob cannot tell the difference between samples in D_{tr} that are marked or not, then Bob has no way of knowing whether or not Alice has marked the data, nor can Bob efficiently filter the dataset to remove D_A from it.

If we assume that there exist watermarking methods such that i) watermarked outputs have the same quality as non-watermarked samples, ii) watermarked samples are indistinguishable from non-watermarked samples for computationally-bounded discriminators, and such that iii) an observer can readily tell the distributions apart if the observer does possess the watermark secret, then a paraphrasing model equipped with an output watermark may be *the* theoretically optimal marking transformation T_m (see Christ et al. (2024); Piet et al. (2025) for some discussion of these ideas in more context). Such an invisible, "non-distortionary" watermark family might serve as a better instantiation of T_m in the proactive dataset protection methodology we study in this work. However, whether the the implicit "learnability" required for the marking to be detectable in Bob's trained model is in fundamental tension with indistinguishability properties is an open question and a fruitful topic for future research.

5. Conclusion

We find the watermark-based dataset inference approach to be a promising alternative to traditional loss-based DIA methods, but overlooked details in previous studies highlight important avenues for future research. Despite our randomization-based detection test yielding a well-calibrated empirical null, the possibility for elevated single-key readings in any setting presents a previously overlooked deployment challenge. Further, any train- vs. detection-time mismatch in document presentation (aligned/packed) can reduce test significance values. Finally, the reading-mode detection procedure from prior work that is required for signal at high dilution may itself be impractical depending on the API access the data owner has against the suspect target model.

In head-to-head comparisons under a controlled experimental design, the loss-based baselines are more performant as DIA methods under an ROC-AUC evaluation. However, by definition they do not admit p-values for assumption-free and calibration-free FPR estimation and they assume greater access to the target model (loss values), some known non-member, distributionally-matched calibration samples, and/or a pool of trained reference models (Table 8). Therefore, which approach is ultimately more useful in practice comes down to which set of assumptions is more realistic for the particular data owner and what level of sensitivity under dilution their use-case requires.

Acknowledgments

This work was supported by DARPA TIAMAT, the NSF TRAILS Institute (2229885), and Coefficient Giving. Computing resources for this project were supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under the Advancements in AI for Science program.

Prepared by LLNL under Contract DE-AC52-07NA27344 and supported by the LLNL-LDRD Program under Projects No. 24-ERD-010, and 24-ERD-058 (LLNL-CONF-2019479). This manuscript has been authored by Lawrence Livermore National Security, LLC under Contract No. DE-AC52-07NA27344 with the U.S. Department of Energy. The United States Government retains, and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

AI Use: Models from both OpenAI (Codex) and Anthropic (Claude) were used extensively in the implementation, experiment management, and paper editing aspects of this work.

References

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Christ, M., Gunn, S., and Zamir, O. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 1125–1139. PMLR, 2024. URL https://arxiv.org/abs/2306.09194.

Cooper, A. F., Gokaslan, A., Ahmed, A., Cyphert, A. B., De Sa, C., Lemley, M. A., Ho, D. E., and Liang, P. Extracting memorized pieces of (copyrighted) books from open-weight language models. *arXiv preprint arXiv:2505.12546*, 2025.

Duan, M., Suri, A., Mireshghallah, N., Min, S., Shi, W., Zettlemoyer, L., Tsvetkov, Y., Choi, Y., Evans, D., and Hajishirzi, H. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*, 2024.

Fernandez, P., Sander, T., Elsahar, H., Chang, H., Souček, T., Lacatusu, V., Tran, T., Rebuffi, S.-A., and Mourachko, A. How good is post-hoc watermarking with language model rephrasing? 2025.

Hayes, J., Shumailov, I., Choquette-Choo, C. A., Jagielski, M., Kaissis, G., Nasr, M., Ghalebikesabi, S., Annamalai, M. S. M. S., Mireshghallah, N., Shilov, I., et al. Exploring the limits of strong membership inference attacks on large language models. *arXiv preprint arXiv:2505.18773*, 2025.

Ilyas, A., Park, S. M., Engstrom, L., Leclerc, G., and Madry, A. Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*, 2022. URL https://arxiv.org/abs/2202.00622.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A Watermark for Large Language Models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://arxiv.org/abs/2301.10226.

Kirchenbauer, J., Mongkolsupawan, J., Wen, Y., Goldstein, T., and Ippolito, D. A fictional q&a dataset for studying memorization and knowledge acquisition. *arXiv preprint arXiv:2506.05639*, 2025.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017. URL https://arxiv.org/abs/1703.04730.

Liu, K. Z., Choquette-Choo, C. A., Jagielski, M., Kairouz, P., Koyejo, S., Liang, P., and Papernot, N. Language models may verbatim complete text they were not explicitly trained on. *arXiv preprint arXiv:2503.17514*, 2025.

Olmo, T., Ettinger, A., Bertsch, A., Kuehl, B., Graham, D., Heineman, D., Groeneveld, D., Brahman, F., Timbers, F., Ivison, H., Morrison, J., Poznanski, J., Lo, K., Soldaini, L., Jordan, M., Chen, M., Noukhovitch, M., Lambert, N., Walsh, P., Dasigi, P., Berry, R., Malik, S., Shah, S., Geng, S., Arora, S., Gupta, S., Anderson, T., Xiao, T., Murray, T., Romero, T., Graf, V., Asai, A., Bhagia, A., Wettig, A., Liu, A., Rangapur, A., Anastasiades, C., Huang, C., Schwenk, D., Trivedi, H., Magnusson, I., Lochner, J., Liu, J., Miranda, L. J. V., Sap, M., Morgan, M., Schmitz, M., Guerquin, M., Wilson, M., Huff, R., Bras, R. L., Xin, R., Shao, R., Skjonsberg, S., Shen, S. Z., Li, S. S., Wilde, T., Pyatkin, V., Merrill, W., Chang, Y., Gu, Y., Zeng, Z., Sabharwal, A., Zettlemoyer, L., Koh, P. W., Farhadi, A., Smith, N. A., and Hajishirzi, H. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Park, S. M., Georgiev, K., Ilyas, A., Leclerc, G., and Madry, A. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023. URL https://arxiv.org/abs/2303.14186.

Piet, J., Sitawarin, C., Fang, V., Mu, N., and Wagner, D. MARKMyWORDS: Analyzing and evaluating language model watermarks. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 68–91. IEEE, 2025. URL https://arxiv.org/abs/2312.00273.

Sander, T., Fernandez, P., Durmus, A., Douze, M., and Furon, T. Watermarking makes language models radioactive. *Advances in Neural Information Processing Systems*, 37:21079–21113, 2024. URL https://arxiv.org/abs/2402.14904.

Sander, T., Fernandez, P., Mahloujifar, S., Durmus, A., and Guo, C. Detecting benchmark contamination through watermarking. *arXiv preprint arXiv:2502.17259*, 2025. URL https://arxiv.org/abs/2502.17259.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. Detecting pretraining data from large language models. In *International Conference on Learning Representations*, volume 2024, pp. 51826–51843, 2024.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Xu, Y. E., Kirchenbauer, J., Savani, Y., Trockman, A., Robey, A., Goldstein, T., Fang, F., and Kolter, J. Z. Antidistillation fingerprinting. *arXiv preprint arXiv:2602.03812*, 2026.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

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

A. Extended Methodological Details

Folded Data Design. We instantiate our folded design under two complementary support regimes that share the same (F, E) axes. The primary, more realistic regime is the *event-split* setting: multiple keyed folds are mixed into each model's training data with event-level groupings preserved within each fold and with the total watermarked share of the training tokens held fixed across F. As F grows, each individual key's per-fold footprint shrinks while the number of distinct keys present grows, simulating a deployment scenario where multiple data owners protect their own subsets with distinct keys inside a single training corpus. The simpler ablation regime is the *simple per-key support* (SKS) setting: each watermarked model trains on exactly one watermarked fold, which isolates the watermarked-fraction-versus-repetition question without sibling-key interference but is unrealistic in that no real deployed corpus would protect only one fold of one data owner's content under a single key. We treat the SKS setting as a structurally clean warm-up that anchors the more realistic event-split readout.

Efficiency-Focused Design Choices. The point of this folded design is to realize a large number of dataset inference trials simulating the proactive watermarking approach as well as running the loss-based methods, all while still controlling computational cost. As a result, certain independence assumptions are necessarily violated. First, the raw source data used at each F level is the same fixed pool of FictionalQA documents, which makes the different experiments correlated with respect to this domain. Second, in the event-split regime each model trains on more than one fold, so a dataset inference experiment for that model with respect to fold i is not fully independent of the experiment testing the same model with respect to fold j which it was also trained on.

However, some of the efficiency-focused choices also help control other confounds. As we increase E, the fold partitions (actual selected documents) and watermark keys used remain identical, making effective epochs the only variable along that axis. Similarly, the folds that the paired clean models train on are the same underlying documents as the ones that the watermarked models train on (just before they were watermarked), controlling for natural variation in modeling difficulty between the watermarked and clean model pairs. In the event-split regime, the models are also trained on different watermarking keys at the same time, simulating a realistic aspect of the deployment scenario where different sub-datasets may be protected with different watermark keys within a single training corpus.

B. Extended Experimental Setup Details

Watermark Scheme Details All four experimental families (event-split finetuning, SKS finetuning, CPT pretraining, and from-scratch pretraining) share the same KGW-style watermark configuration: greenlist fraction $\gamma = 0.5$, logit bias $\delta = 4.0$, and a 2-token prefix used to seed the PRF that selects each step's greenlist. Headline significance values come from the empirical-null permutation test of Section 2.2.

Detection is run in teacher-forced *reading* mode: the target model is run over the tokenized source text and scored at each position under the preceding tokens of that same source text, not on free generations. We score under argmax mode—counting positions where the model's greedy next-token lands inside the keyed greenlist. Repeated prefix windows are deduplicated by context so that high-frequency local contexts cannot dominate either the watermark score or the empirical null. For the empirical null, we draw $M = 10^6$ randomized keys per surface from a fixed integer keyspace; this gives an exact p-value resolution of 10^{-6}, with an empirical-Gaussian fit to the same null distribution available as an extrapolated reference where smaller p-values are needed.

The watermark keys themselves are scaffolded differently across the three experimental settings. The event-split finetuning grid uses, at each F level, a fresh set of F fold-specific keys (one per fold), and the detector key for a given watermark surface is always matched to the fold ID being read. The SKS ablation does not use fold-specific keys; it uses four fixed keys, each held constant across the entire (F, E) grid for one of four model families, with the detector key matched to the family's single key. Phase 2 pretraining (both CPT and from-scratch) reuses the same two $F = 2$ fold keys from the Phase 1 event-split anchor row across all ten schedules and both initialization regimes; this is deliberate cross-phase calibration, which is also why the warm-key handedness present in the Phase 1 $F = 2$ scaffold (Section C.7) is visible in the Phase 2 keyed readouts.

Data Folding. For each finetuning experiment, we define a fold factor F and partition the FictionalQA documents into F equal folds. We then train a small population of models per cell with paired clean and watermarked twins, so that each watermarked positive has a matched clean "false-probe" negative trained on the unmarked version of the same documents.

To simulate the watermark-based dataset protection approach, for every fold D_i, we create a paraphrased version of the documents using a paraphraser model running the watermark decoding scheme with key k_i, producing $D_{k_i} = T_m(D_i, k_i)$. The watermarked model $f^{wm}_{\theta_i}$ is trained on a mixture that includes D_{k_i} and other data, while the corresponding clean twin $f^{c}_{\theta_i}$ trains on D_i in its original form before the paraphrasing transformation. We also vary the number of effective epochs E over the watermarked subset within a run while holding the per-cell training-token budget fixed across (F, E); samples from the base data source are not repeated. Table 1 summarizes the resulting effective per-key exposure across the grid; the event-split and SKS regimes share this idealized table.

C. Finetuning Event-Split Exhaustive Readout

This appendix carries the full per-cell readout of the event-split finetuning grid that the main body's Figure 2 introduces. The main body uses the empirical-exact reference for both the keyed-signal and DIA heatmaps; the subsections below add the empirical-Gaussian companion of each, the packed-surface counterparts of the keyed/null pair, the cross-key sham-null heatmaps, the realized exposure and $\hat{E}$ tables, the row-level MIA table for the baselines, and the null-validity panel.

C.1. Empirical-Gaussian and Packed-Surface Heatmap Companions

Figures 2 and 7 to 9 carry the four event-split keyed/null pairs across the (aligned, packed) surface and (exact, Gaussian) p-value-type combinations; the aligned-exact keyed half is also shown in the main-body Figure 2. Figures 3 and 10 carry the matching watermark whole-model DIA AUC pairs, with aligned and packed surfaces shown side-by-side under each p-value type.

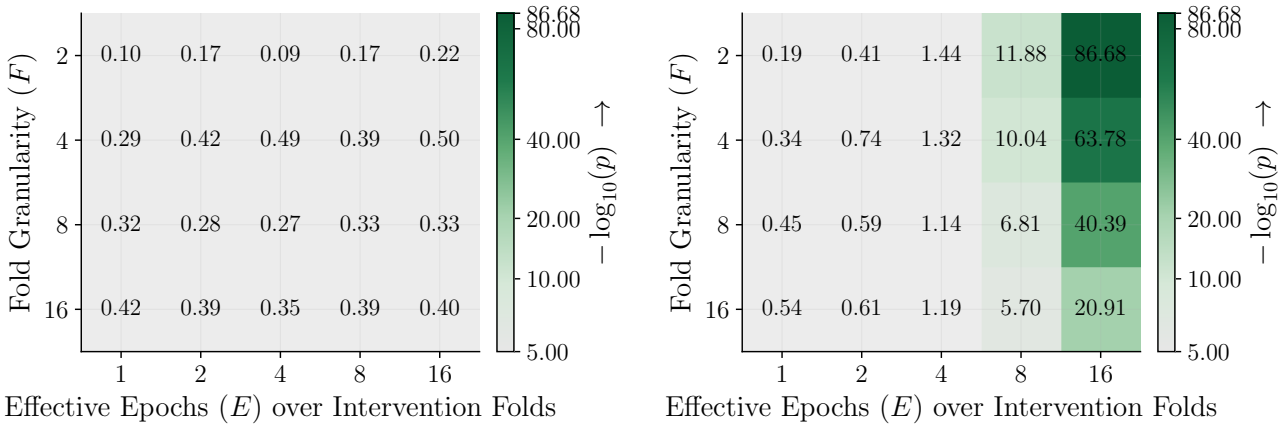

Figure 7. Finetuning event-split matched clean-twin false-probe null (left) and keyed signal (right) across the $F \times E$ grid on the aligned unpacked detection surface, scored as $-\log_{10} p$ under the empirical-Gaussian reference.

C.2. Event-Split Exposure Trend Curves

Figure 11 is a re-visualization of the same keyed-signal data shown in the right panel of Figure 2, but using the empirical Gaussian extrapolation p-values from Figure 7 and re-cast on a continuous exposure axis with separate lines per F to make the exposure trend and the cross-F separation easier to read at a glance. At low exposure the per-key support is too small for keyed signal to lift off in any of the F rows, while at high exposure the curves separate cleanly with F.

C.3. Cross-Key Sham-Null Heatmaps

Figures 12 and 13 carry the cross-key sham-null heatmaps for the event-split grid, where the watermark detector is queried with a key the target model never saw in training. These act as an additional negative control beyond the matched clean-twin false-probe null, isolating per-key idiosyncrasy of the detection surface from the matched-pair design.

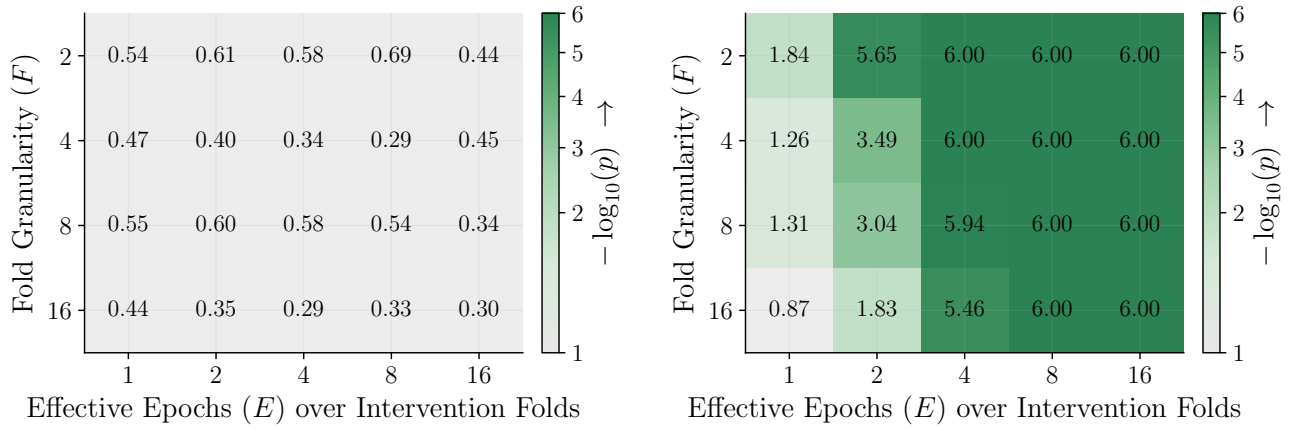

Figure 8. Finetuning event-split matched clean-twin false-probe null (left) and keyed signal (right) across the $F \times E$ grid on the packed detection surface, scored as $-\log_{10} p$ under the empirical-exact null. The packed surface is a more permissive oracle baseline than the aligned surface.

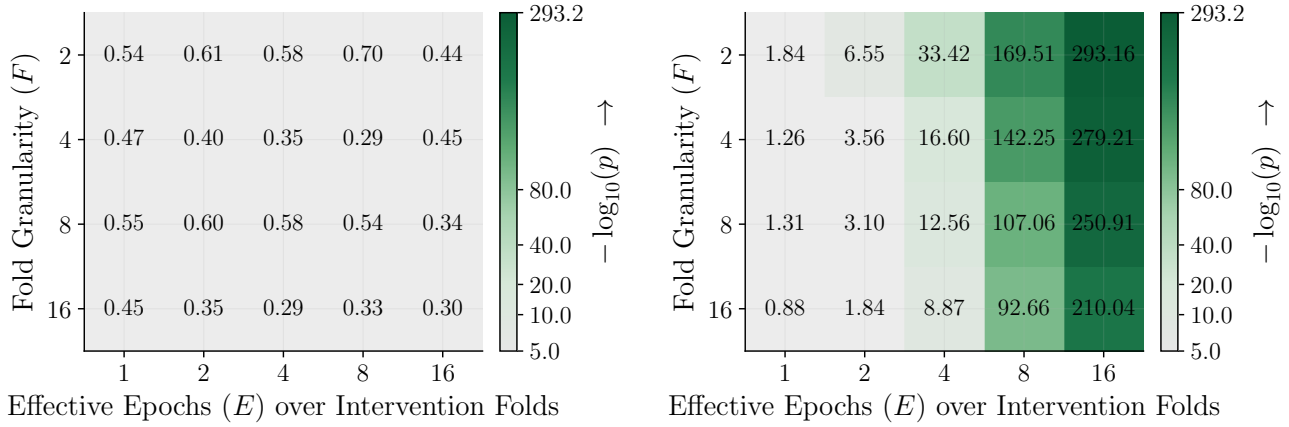

Figure 9. Finetuning event-split matched clean-twin false-probe null (left) and keyed signal (right) across the $F \times E$ grid on the packed detection surface, scored as $-\log_{10} p$ under the empirical-Gaussian reference.

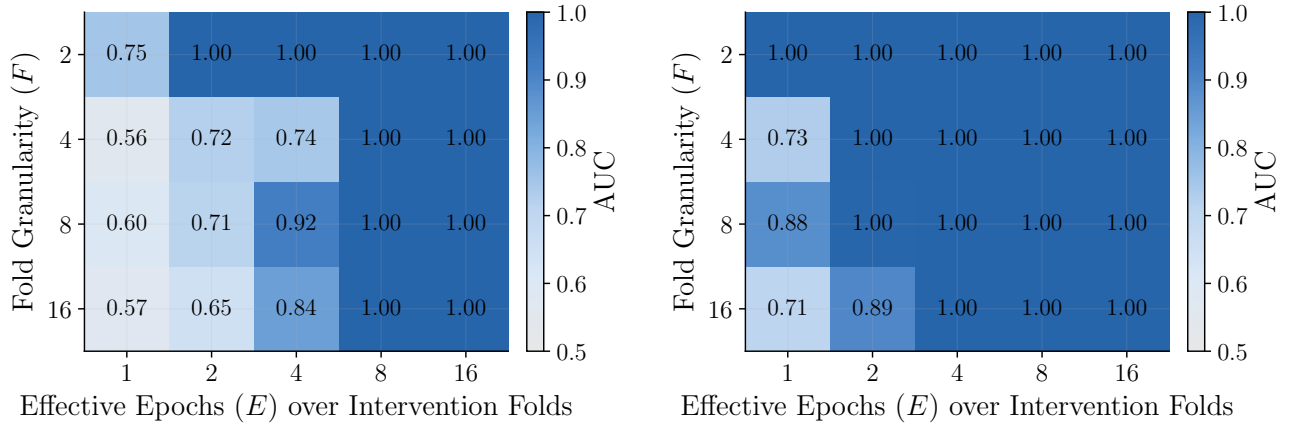

Figure 10. Finetuning event-split watermark whole-model DIA AUC across the $F \times E$ grid, on the aligned (left) and packed (right) detection surfaces, both scored against an empirical-Gaussian null.

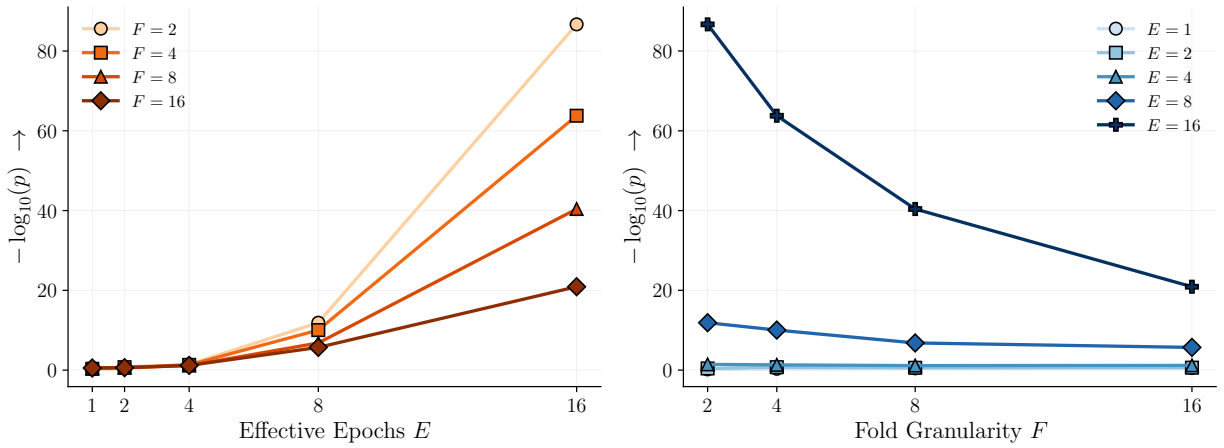

Figure 11. Finetuning event-split keyed-signal exposure response across the grid, tracing $-\log_{10} p$ as a function of effective per-key exposure with separate lines for each fold count F. This figure plots the same per-cell keyed-signal values as the right panel of Figure 2, just computed with the empirical Gaussian extrapolated p-values from Figure 7 and re-cast on a continuous exposure axis to make the trend clearer.

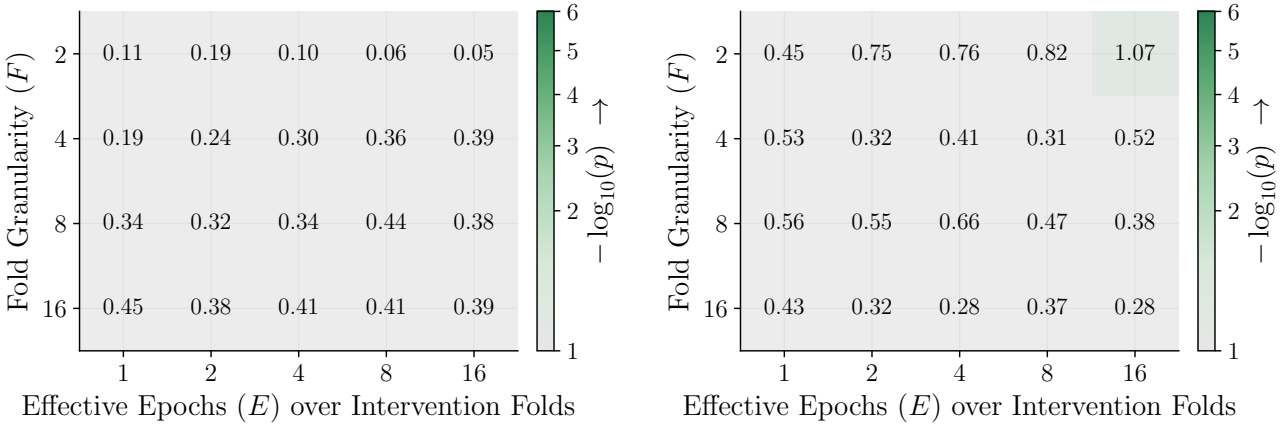

Figure 12. Finetuning event-split cross-key sham-null heatmap across the $F \times E$ grid, on the aligned (left) and packed (right) detection surfaces, scored as $-\log_{10} p$ under the empirical-exact null. The watermark detector is queried with a key the target model never saw in training, providing a negative control beyond the matched clean-twin false-probe null.

C.4. Event-Split Realized Exposure and $\hat{E}$ Per-Cell Readbacks

Table 3 reports the realized normalized exposure $\hat{E}/F$ per cell of the event-split finetuning grid. Table 4 reports the underlying realized $\hat{E}$ values. The corresponding idealized epoch counts E are the column headers of Table 1, which the event-split construction shares with SKS.

Table 3. Event-split finetuning: realized normalized exposure summary ($\hat{E}/F$).

	$E = 1$	$E = 2$	$E = 4$	$E = 8$	$E = 16$
$F = 2$	0.5612	1.1428	2.2880	4.3259	8.2775
$F = 4$	0.2944	0.5955	1.1603	2.2555	4.3436
$F = 8$	0.1490	0.2996	0.5883	1.1481	2.2133
$F = 16$	0.0837	0.1699	0.3310	0.6457	1.2488

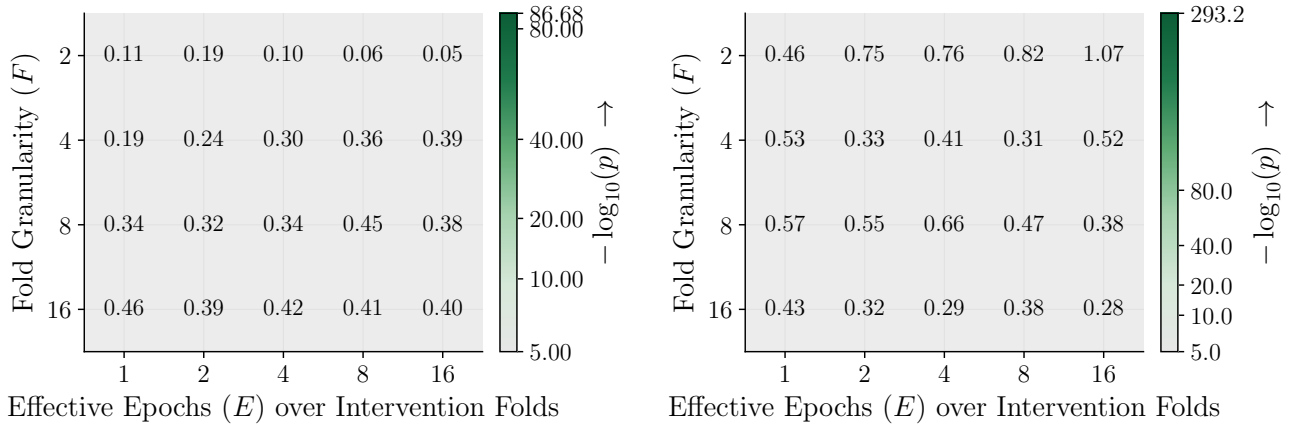

Figure 13. Finetuning event-split cross-key sham-null heatmap across the $F \times E$ grid, on the aligned (left) and packed (right) detection surfaces, scored as $-\log_{10} p$ under the empirical-Gaussian null.

Table 4. Event-split finetuning: realized exposure summary ($\hat{E}$).

	$E = 1$	$E = 2$	$E = 4$	$E = 8$	$E = 16$
$F = 2$	1.1223	2.2857	4.5760	8.6519	16.5550
$F = 4$	1.1776	2.3818	4.6412	9.0218	17.3744
$F = 8$	1.1922	2.3970	4.7062	9.1846	17.7066
$F = 16$	1.3396	2.7188	5.2960	10.3307	19.9814

C.5. Event-Split Training Scale and Trial Geometry

Table 5 reports the per-cell watermarked-token totals (mean, min–max across paired models) and the corresponding fraction of each model's 131M-token training budget. Table 6 reports the per-cell paired-model counts and the resulting n_+/n_- trial counts that drive each cell's whole-model DIA AUC; both are summarized in Section 3.1.

C.6. Event-Split Loss-Based and Reference-Model Baselines Row-Level MIA

C.7. Event-Split Null-Validity Extended

The clipped chart Figure 4 in the main body, equivalent to the left side of Figure 14 confirms that once pooled across many distinct positive keys, the empirical exact null is close to uniform at the 1M-null scale and respects standard tail-rate thresholds, supporting the use of the empirical-null permutation test as the headline reference distribution for the keyed readout. Per-key idiosyncrasy is a separate concern visible in the right-hand trace: at very small fold counts a single key can run systematically warm even when the pooled null is well-calibrated, and the inherited $F = 2$ scaffold reused in pretraining (Section 3.2) carries one such warm key.

D. Finetuning SKS Exhaustive Readout

This appendix reports the simple per-key support (SKS) ablation, where each model trains on exactly one watermarked fold so the per-model fictional support fraction shrinks as $1/F$ instead of being held fixed across F. SKS is structurally simpler than the event-split regime (Section C) but unrealistic in that no real deployed corpus would protect only one fold of one data owner's content under a single key. The same idealized per-key exposure (Table 1) governs both regimes; what differs is sibling support and the per-cell positive/negative trial budget (Table 12). We use SKS as a watermark-only sanity check on the per-key scaling story without sibling-key interference, and run the loss-based and reference-model comparison only on the more realistic event-split regime where the row-level baselines are meaningful.

Table 5. Event-split finetuning: training scale context. Per-cell watermark token totals are relative to 131.1M train tokens per run. The percent columns report watermark-token share of total train tokens.

Cell	Target (E/F)	WM tokens mean	WM tokens min-max	Mean %	Range %
(F2,E1)	0.5000	0.5633M	0.5367M-0.59M	0.430%	0.409%-0.450%
(F2,E2)	1.0000	1.143M	1.082M-1.205M	0.872%	0.825%-0.919%
(F2,E4)	2.0000	2.288M	2.171M-2.405M	1.746%	1.657%-1.835%
(F2,E8)	4.0000	4.318M	4.294M-4.343M	3.295%	3.276%-3.313%
(F2,E16)	8.0000	8.294M	8.239M-8.35M	6.328%	6.286%-6.370%
(F4,E1)	0.2500	0.2875M	0.2212M-0.3687M	0.219%	0.169%-0.281%
(F4,E2)	0.5000	0.5797M	0.463M-0.6883M	0.442%	0.353%-0.525%
(F4,E4)	1.0000	1.122M	1.028M-1.307M	0.856%	0.785%-0.997%
(F4,E8)	2.0000	2.181M	1.979M-2.339M	1.664%	1.510%-1.785%
(F4,E16)	4.0000	4.2M	3.802M-4.49M	3.204%	2.901%-3.426%
(F8,E1)	0.1250	0.1417M	0.09013M-0.2008M	0.108%	0.069%-0.153%
(F8,E2)	0.2500	0.286M	0.2171M-0.3728M	0.218%	0.166%-0.284%
(F8,E4)	0.5000	0.5587M	0.463M-0.7088M	0.426%	0.353%-0.541%
(F8,E8)	1.0000	1.09M	0.9177M-1.397M	0.832%	0.700%-1.066%
(F8,E16)	2.0000	2.103M	1.758M-2.454M	1.604%	1.341%-1.872%
(F16,E1)	0.0625	0.07084M	0.02868M-0.1311M	0.054%	0.022%-0.100%
(F16,E2)	0.1250	0.143M	0.09013M-0.2089M	0.109%	0.069%-0.159%
(F16,E4)	0.2500	0.2794M	0.2212M-0.3605M	0.213%	0.169%-0.275%
(F16,E8)	0.5000	0.5452M	0.4466M-0.7293M	0.416%	0.341%-0.556%
(F16,E16)	1.0000	1.051M	0.8153M-1.377M	0.802%	0.622%-1.050%

Table 6. Event-split finetuning: model and watermark DIA trial geometry. The WM and clean model columns count target models per cell, and the WM DIA trial column counts the positive/negative pooled trials used for the watermark DIA AUC.

Cell	WM models n_+	Clean models n_-	WM DIA trials n_+/n_-
(F2,E1)	2	2	2 / 2
(F2,E2)	2	2	2 / 2
(F2,E4)	2	2	2 / 2
(F2,E8)	2	2	2 / 2
(F2,E16)	2	2	2 / 2
(F4,E1)	6	6	12 / 12
(F4,E2)	6	6	12 / 12
(F4,E4)	6	6	12 / 12
(F4,E8)	6	6	12 / 12
(F4,E16)	6	6	12 / 12
(F8,E1)	12	12	48 / 48
(F8,E2)	12	12	48 / 48
(F8,E4)	12	12	48 / 48
(F8,E8)	12	12	48 / 48
(F8,E16)	12	12	48 / 48
(F16,E1)	12	12	96 / 96
(F16,E2)	12	12	96 / 96
(F16,E4)	12	12	96 / 96
(F16,E8)	12	12	96 / 96
(F16,E16)	12	12	96 / 96

D.1. SKS Empirical-Gaussian and Packed-Surface Heatmap Companions

Figures 15 to 20 carry the SKS keyed/null pair and DIA AUC heatmap pair on the aligned-exact, aligned-Gaussian, packed-exact, and packed-Gaussian detection surfaces.

Table 7. Event-split finetuning: row-level MIA AUC comparison. Entries marked N/A indicate statistics that are not estimable in the available cell geometry; for example, LiRA in F=2 cells lacks sufficient in-reference models.

	Watermark Readout	Loss-based Row MIA			Ref-model Row MIA		
Cell	$\mathrm{WM} - \log_{10}(p)$	Raw-loss	Argmax	min-k_{10}	rMIA-simple	rMIA	LiRA
(F2,E1)	0.1894	0.6096	0.6083	0.6292	1.0000	0.9993	N/A
(F2,E2)	0.4135	0.6988	0.6987	0.7318	1.0000	0.9993	N/A
(F2,E4)	1.4385	0.8521	0.8508	0.8991	1.0000	0.9993	N/A
(F2,E8)	11.8808	0.9744	0.9744	0.9707	0.9999	0.9993	N/A
(F2,E16)	86.6829	0.9832	0.9870	0.9683	0.9998	0.9993	N/A
(F4,E1)	0.3363	0.6041	0.6013	0.6237	0.9760	0.9805	0.8050
(F4,E2)	0.7380	0.6901	0.6896	0.7229	0.9985	0.9978	0.9165
(F4,E4)	1.3158	0.8360	0.8336	0.8813	1.0000	0.9993	0.9415
(F4,E8)	10.0393	0.9711	0.9691	0.9718	1.0000	0.9993	0.9416
(F4,E16)	63.7768	0.9745	0.9788	0.9608	0.9997	0.9992	0.9707
(F8,E1)	0.4487	0.6022	0.6010	0.6207	0.9645	0.9686	0.9668
(F8,E2)	0.5877	0.6852	0.6849	0.7163	0.9952	0.9946	0.9981
(F8,E4)	1.1352	0.8306	0.8284	0.8731	0.9992	0.9986	0.9996
(F8,E8)	6.8089	0.9564	0.9542	0.9552	0.9996	0.9990	0.9992
(F8,E16)	40.3926	0.9672	0.9717	0.9530	0.9995	0.9990	0.9991
(F16,E1)	0.5373	0.5975	0.5965	0.6157	0.9302	0.9341	0.9486
(F16,E2)	0.6138	0.6774	0.6758	0.7068	0.9771	0.9768	0.9911
(F16,E4)	1.1855	0.8148	0.8098	0.8471	0.9904	0.9897	0.9926
(F16,E8)	5.6955	0.9148	0.9166	0.9064	0.9903	0.9917	0.9898
(F16,E16)	20.9064	0.9294	0.9355	0.9132	0.9959	0.9962	0.9940

Table 8. To contextualize the performance differences between the watermark-based DIA technique and other baselines presented in Table 2, here we summarize the key assumptions about the data owner's (i.e. attacker's) control over the data and access to the target model needed to make a dataset membership claim. **Qualifications:** [1]The "reading-mode" method of Sander et al. (2024; 2025) requires the ability to check what the next-token prediction of the target model would be conditioned on a prefix of the protected sample. While in theory this could be accomplished by repeatedly querying the model while iteratively appending tokens to the prompt (expensive), the practicality of this operation given modern frontier model APIs is questionable.[2]We mark this row as such to draw attention to the fact that performing academic analyses centering ROC-AUC allows us to ignore the necessity of calibrating a single decision threshold to deploy. However, in practice, for all MIA/DIA techniques that do not admit a p-value or other analytic (FP) error estimate, held-out validation samples must be used to calibrate the decision threshold.

Access Assumption	Watermarked	Raw-loss	Argmax	min-k_{10}	rMIA-simple	rMIA	LiRA
Data owner can modify their data before it is included in the training dataset.	✓						
Data owner has next-token prediction[1] access to target model.	✓		✓				
Data owner must maintain held-out pool of additional samples for calibration[2].		✓	✓	✓	✓	✓	✓
Data owner has loss or logit access to target model.		✓		✓	✓	✓	✓
Data owner has compute and target model details needed to train reference models.					✓	✓	✓

D.2. SKS Exposure Trend Curves

Figure 21 is a re-visualization of the same keyed-signal data shown in the right panel of Figure 15, re-cast on a continuous exposure axis with separate lines per F to make the exposure trend and the cross-F separation easier to read at a glance: at

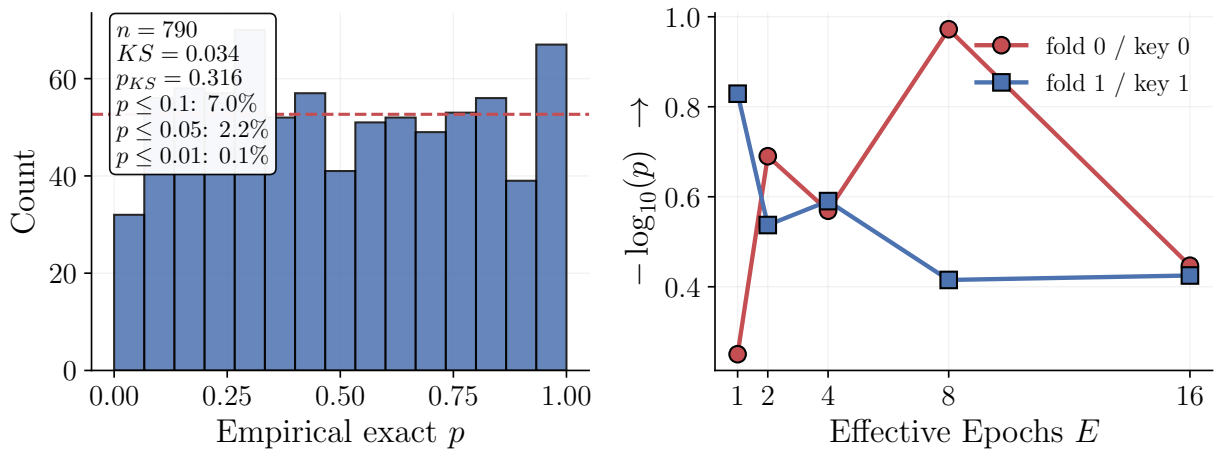

Figure 14. Event-grid null-validity panel. **Left:** histogram of pooled empirical exact p-values from the packed matched-clean-negative whole-model readings across the (F, E) event grid. The annotated statistic and p-value are from a one-sample Kolmogorov-Smirnov test of these pooled p-values against $\mathrm{Uniform}(0, 1)$; the dashed horizontal line is the expected per-bin count under a uniform histogram with the plotted binning. **Right:** $-\log_{10} p$ trace of the $F = 2$ warm-key slice of the same packed null family plotted against E, shown for context only and not part of the KS test on the left.

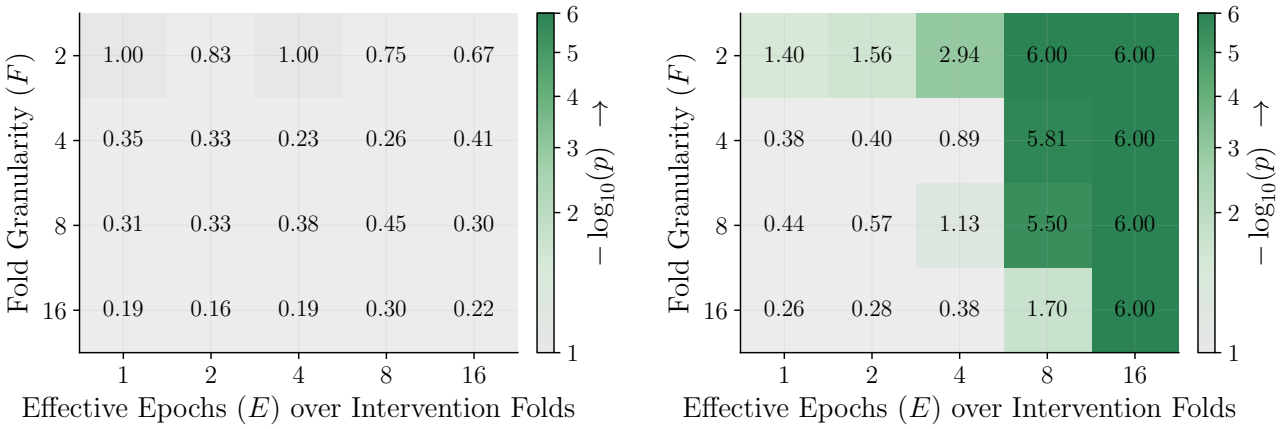

Figure 15. Finetuning SKS matched clean-twin false-probe null (left) and keyed signal (right) across the $F \times E$ grid on the aligned unpacked detection surface, scored as $-\log_{10} p$ under an empirical-exact reference. The false-probe null stays quiet on the same surface, validating the matched clean-twin negative as the right baseline against which to read the keyed map; the keyed signal grows monotonically with E and decays with F as the per-key support fraction $1/F$ shrinks.

low exposure the per-key support is too small for keyed signal to lift off in any of the F rows, while at high exposure the curves separate cleanly with F.

D.3. SKS Realized Exposure and $\hat{E}$ Per-Cell Readbacks

Table 9 reports the realized normalized exposure $\hat{E}/F$ per cell of the finetuning SKS grid, the direct realized counterpart to the idealized E/F values in Table 1. Table 10 reports the underlying realized $\hat{E}$ values. The discrepancy between idealized and realized values here is the realized overshoot of the watermarked subset's effective epoch count relative to the planned schedule.

D.4. SKS Training Scale and Trial Geometry

Table 11 reports the per-cell watermarked-token totals (mean, min–max across paired models) and the corresponding fraction of each model's 131M-token training budget. Table 12 reports the per-cell paired-model counts and the resulting

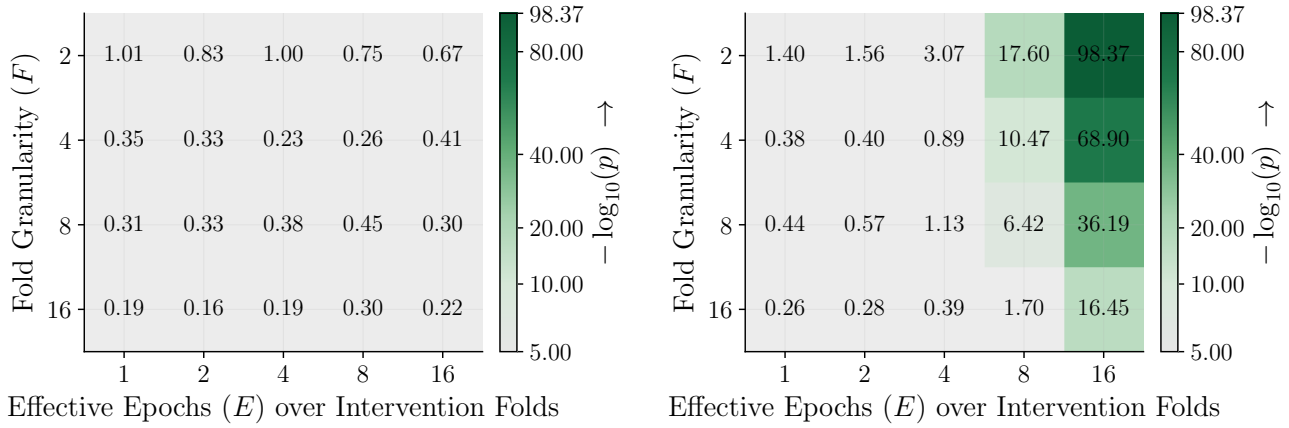

Figure 16. Finetuning SKS matched clean-twin false-probe null (left) and keyed signal (right) across the $F \times E$ grid on the aligned unpacked detection surface, scored as $-\log_{10} p$ under the empirical-Gaussian reference.

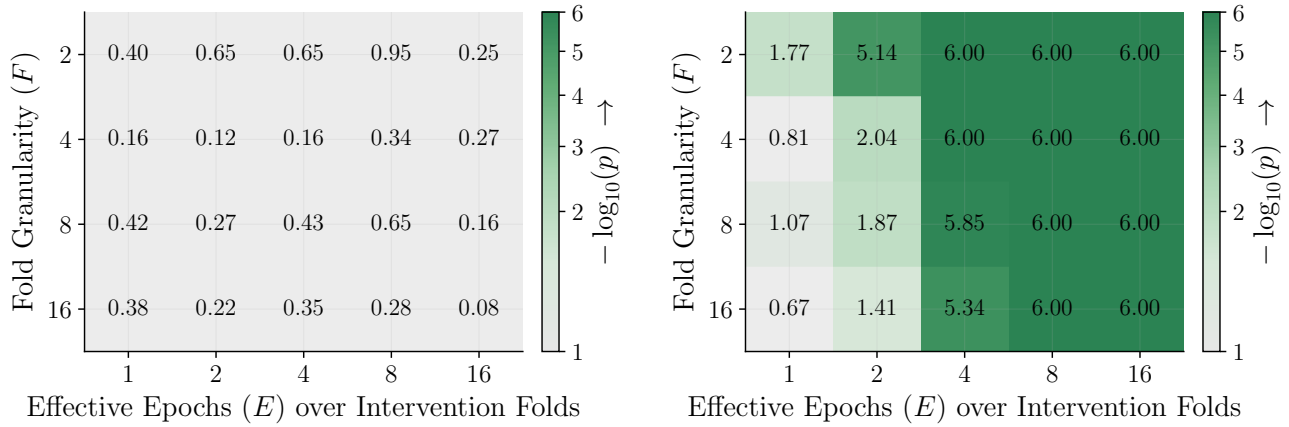

Figure 17. Finetuning SKS matched clean-twin false-probe null (left) and keyed signal (right) across the $F \times E$ grid on the packed detection surface, scored as $-\log_{10} p$ under the empirical-exact null. The packed surface is a more permissive oracle baseline than the aligned surface.

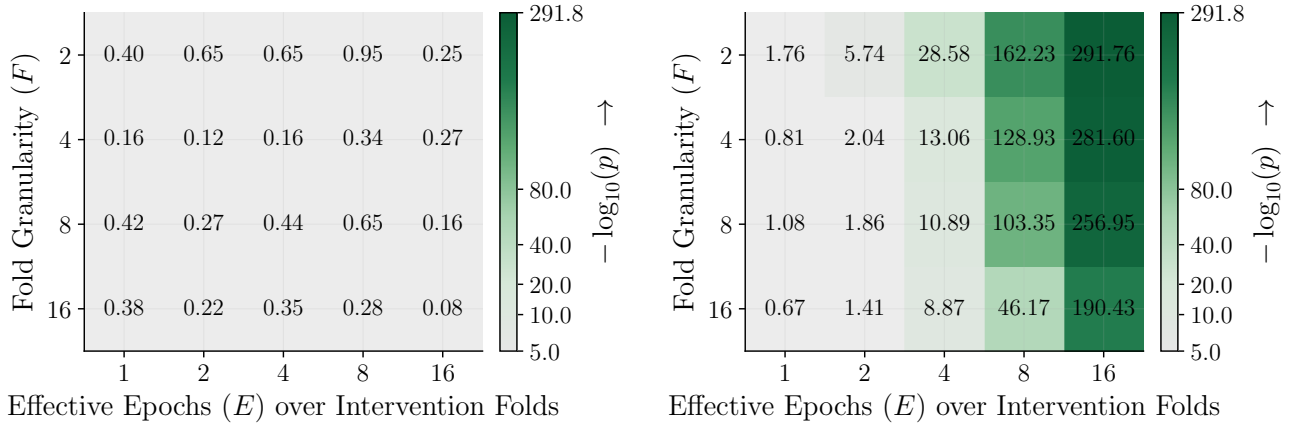

Figure 18. Finetuning SKS matched clean-twin false-probe null (left) and keyed signal (right) across the $F \times E$ grid on the packed detection surface, scored as $-\log_{10} p$ under the empirical-Gaussian reference.

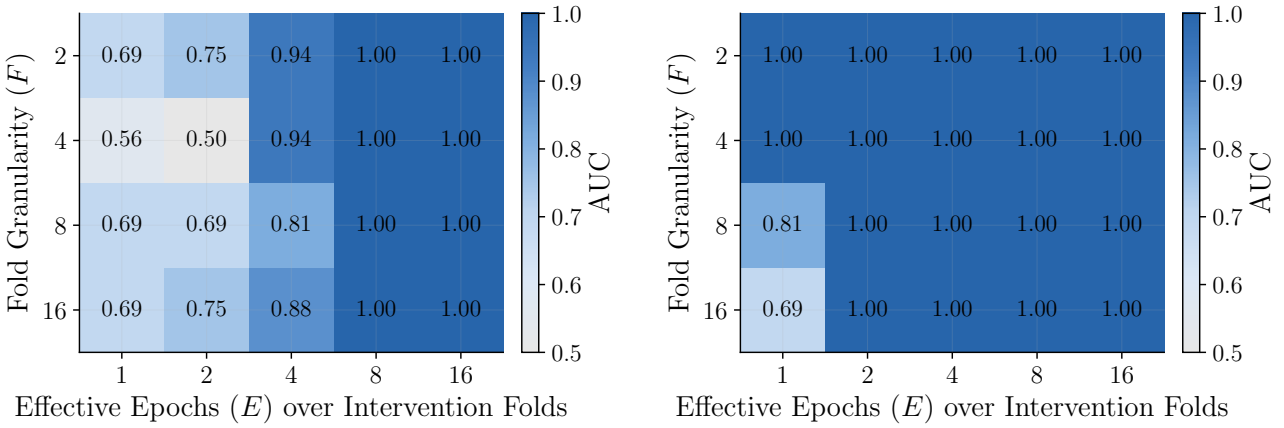

Figure 19. Finetuning SKS watermark whole-model DIA AUC across the $F \times E$ grid, on the aligned (left) and packed (right) detection surfaces, both scored against an empirical-exact null. On the aligned surface the AUC saturates at 1.0 from $E = 8$ onward at every F, while the lowest-E corner remains coarse with only eight trials per cell. On the packed surface, the more permissive oracle recovers several of those low-exposure cells, reaching 1.0 one to two E-steps earlier across the grid.

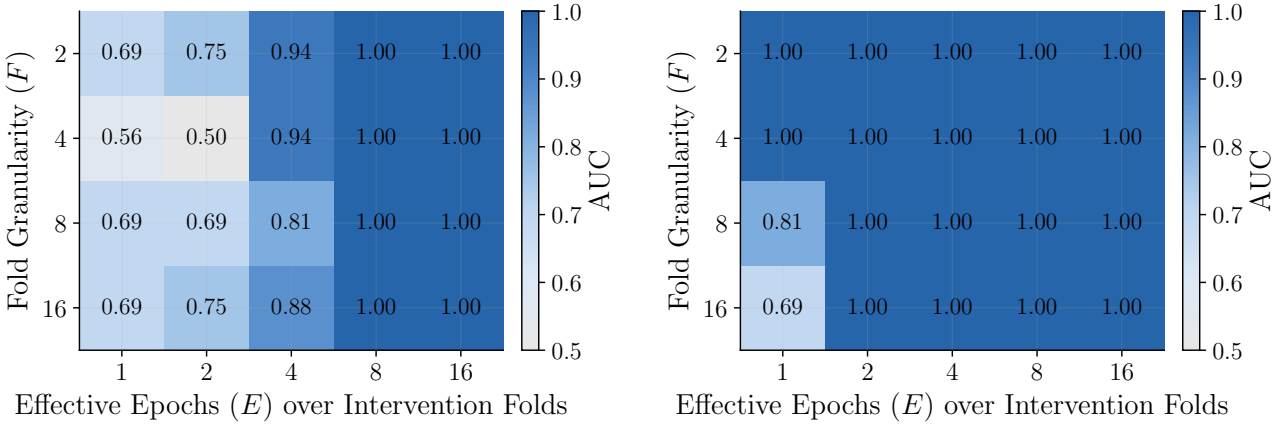

Figure 20. Finetuning SKS watermark whole-model DIA AUC across the $F \times E$ grid, on the aligned (left) and packed (right) detection surfaces, both scored against an empirical-Gaussian null.

Table 9. SKS finetuning: realized normalized exposure summary ($\hat{E}/F$).

	$E = 1$	$E = 2$	$E = 4$	$E = 8$	$E = 16$
$F = 2$	0.5557	1.1246	2.2800	4.3204	8.3766
$F = 4$	0.2755	0.5584	1.1609	2.2062	4.3422
$F = 8$	0.1549	0.2884	0.5772	1.1895	2.1984
$F = 16$	0.0911	0.1797	0.3477	0.6667	1.2930

Table 10. SKS finetuning: realized exposure summary ($\hat{E}$).

	$E = 1$	$E = 2$	$E = 4$	$E = 8$	$E = 16$
$F = 2$	1.1114	2.2492	4.5600	8.6408	16.7533
$F = 4$	1.1021	2.2335	4.6437	8.8248	17.3686
$F = 8$	1.2388	2.3070	4.6175	9.5159	17.5870
$F = 16$	1.4583	2.8750	5.5625	10.6667	20.6875

n_+/n_- trial counts; SKS holds the per-cell trial budget fixed at $4/4$ across the grid, since each model trains on exactly one watermarked fold and the per-cell positive/negative budget is the same at every (F, E).

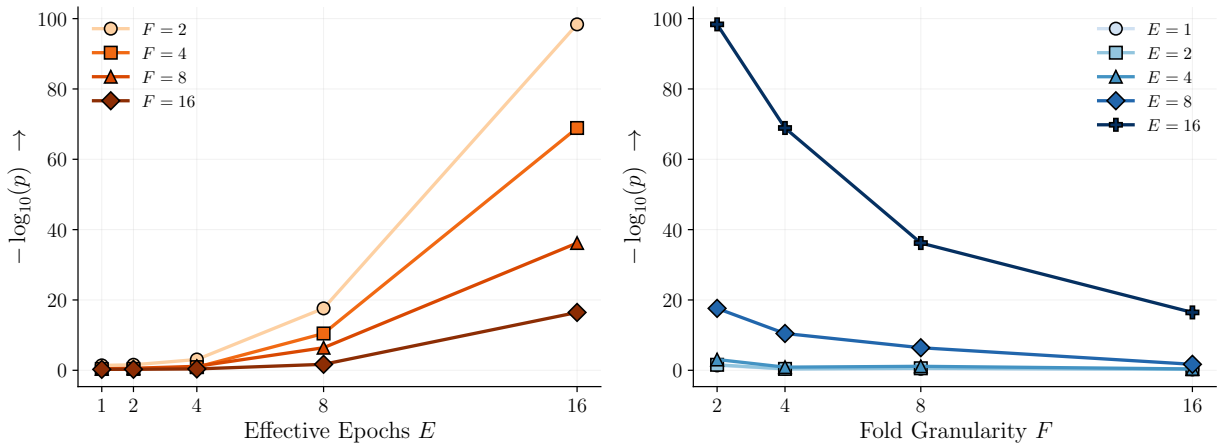

Figure 21. Finetuning SKS keyed-signal exposure response across the grid, tracing $-\log_{10} p$ as a function of effective per-key exposure with separate lines for each fold count F. This figure plots the same per-cell keyed-signal values as the right panel of Figure 15, just re-cast on a continuous exposure axis to make the trend clearer.

Table 11. SKS finetuning: training scale context. Per-cell watermark token totals are relative to 131.1M train tokens per run. The percent columns report watermark-token share of total train tokens.

Cell	Target (E/F)	WM tokens mean	WM tokens min-max	Mean %	Range %
(F2,E1)	0.5000	0.5562M	0.4671M-0.6719M	0.424%	0.356%-0.513%
(F2,E2)	1.0000	1.133M	0.9628M-1.168M	0.865%	0.735%-0.891%
(F2,E4)	2.0000	2.282M	2.13M-2.417M	1.741%	1.625%-1.844%
(F2,E8)	4.0000	4.354M	4.236M-4.449M	3.322%	3.232%-3.395%
(F2,E16)	8.0000	8.425M	8.12M-8.677M	6.428%	6.195%-6.620%
(F4,E1)	0.2500	0.2709M	0.2417M-0.3073M	0.207%	0.184%-0.234%
(F4,E2)	0.5000	0.548M	0.5039M-0.6105M	0.418%	0.384%-0.466%
(F4,E4)	1.0000	1.142M	1.028M-1.25M	0.871%	0.785%-0.953%
(F4,E8)	2.0000	2.165M	2.081M-2.257M	1.652%	1.588%-1.722%
(F4,E16)	4.0000	4.26M	3.974M-4.404M	3.250%	3.032%-3.360%
(F8,E1)	0.1250	0.1541M	0.1393M-0.1762M	0.118%	0.106%-0.134%
(F8,E2)	0.2500	0.2868M	0.2294M-0.3278M	0.219%	0.175%-0.250%
(F8,E4)	0.5000	0.5736M	0.5408M-0.6023M	0.438%	0.413%-0.459%
(F8,E8)	1.0000	1.183M	1.053M-1.25M	0.903%	0.803%-0.953%
(F8,E16)	2.0000	2.204M	2.003M-2.397M	1.681%	1.528%-1.829%
(F16,E1)	0.0625	0.0717M	0.05736M-0.09013M	0.055%	0.044%-0.069%
(F16,E2)	0.1250	0.1413M	0.1024M-0.1762M	0.108%	0.078%-0.134%
(F16,E4)	0.2500	0.2735M	0.2294M-0.3196M	0.209%	0.175%-0.244%
(F16,E8)	0.5000	0.5244M	0.4712M-0.5777M	0.400%	0.359%-0.441%
(F16,E16)	1.0000	1.017M	0.9341M-1.098M	0.776%	0.713%-0.838%

D.5. SKS Watermark-Only DIA Numeric Table

Table 13 carries the underlying numeric AUC values that drive the SKS DIA heatmap pair in Figure 19. We deliberately keep this table watermark-only: the SKS support construction is a clean per-key scaling ablation in which each model trains on exactly one watermarked fold, and the loss-based / reference-model row-level baselines are only meaningful against the realistic event-split regime where multiple keys coexist, so the head-to-head comparison against those baselines is run only there (Section C.6).

Table 12. SKS finetuning: model and watermark DIA trial geometry. The WM and clean model columns count target models per cell, and the WM DIA trial column counts the positive/negative pooled trials used for the watermark DIA AUC.

Cell	WM models n_+	Clean models n_-	WM DIA trials n_+/n_-
(F2,E1)	4	4	4 / 4
(F2,E2)	4	4	4 / 4
(F2,E4)	4	4	4 / 4
(F2,E8)	4	4	4 / 4
(F2,E16)	4	4	4 / 4
(F4,E1)	4	4	4 / 4
(F4,E2)	4	4	4 / 4
(F4,E4)	4	4	4 / 4
(F4,E8)	4	4	4 / 4
(F4,E16)	4	4	4 / 4
(F8,E1)	4	4	4 / 4
(F8,E2)	4	4	4 / 4
(F8,E4)	4	4	4 / 4
(F8,E8)	4	4	4 / 4
(F8,E16)	4	4	4 / 4
(F16,E1)	4	4	4 / 4
(F16,E2)	4	4	4 / 4
(F16,E4)	4	4	4 / 4
(F16,E8)	4	4	4 / 4
(F16,E16)	4	4	4 / 4

Table 13. SKS finetuning: watermark whole-model DIA AUC summary.

Cell	Aligned Exact	Packed Exact
(F2,E1)	0.6875	1.0000
(F2,E2)	0.7500	1.0000
(F2,E4)	0.9375	1.0000
(F2,E8)	1.0000	1.0000
(F2,E16)	1.0000	1.0000
(F4,E1)	0.5625	1.0000
(F4,E2)	0.5000	1.0000
(F4,E4)	0.9375	1.0000
(F4,E8)	1.0000	1.0000
(F4,E16)	1.0000	1.0000
(F8,E1)	0.6875	0.8125
(F8,E2)	0.6875	1.0000
(F8,E4)	0.8125	1.0000
(F8,E8)	1.0000	1.0000
(F8,E16)	1.0000	1.0000
(F16,E1)	0.6875	0.6875
(F16,E2)	0.7500	1.0000
(F16,E4)	0.8750	1.0000
(F16,E8)	1.0000	1.0000
(F16,E16)	1.0000	1.0000

D.6. SKS Null-Validity

Figure 22 confirms that, once pooled across many distinct positive keys, the empirical exact null is close to uniform at the 1M-null scale and respects standard tail-rate thresholds. This holds in both the event-split grid (Figure 4) and in this SKS ablation, supporting the use of the empirical-null permutation test as the headline reference distribution across both regimes.

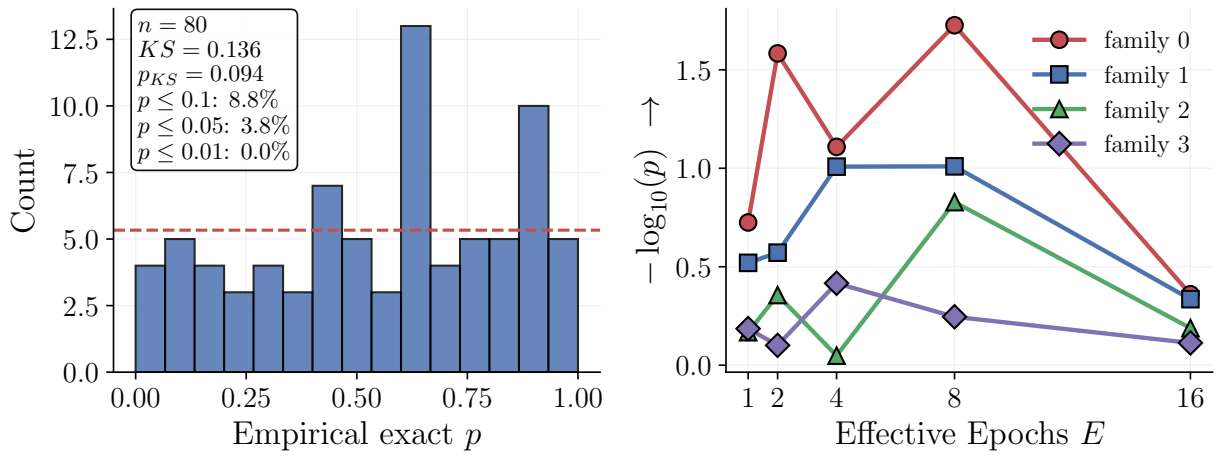

Figure 22. SKS null-validity panel. **Left:** histogram of pooled empirical exact p-values from the packed clean-model watermark-surface false-probe rows across the depth-4 SKS grid. The annotated statistic and p-value are from a one-sample Kolmogorov-Smirnov test of these pooled p-values against $\mathrm{Uniform}(0, 1)$; the dashed horizontal line is the expected per-bin count under a uniform histogram with the plotted binning. **Right:** $-\log_{10} p$ trace of the $F = 2$ warm-key slice of the same packed null family plotted against E, shown for context only and not part of the KS test on the left.

E. Pretraining Exhaustive Readout

This appendix carries the full per-init readout of the pretraining schedule sweep to complement the main body's Figures 5 and 6. The structure is grouped first by initialization regime (CPT then from-scratch), and within each by aligned-then-packed surface, exact-then-Gaussian p-value type, with the watermark whole-model DIA bar pairs trailing each init's keyed/null block. The split row-level MIA and whole-model DIA baseline tables for both initialization regimes are reported below the per-init bar companions, and each whole-model DIA cell is computed over $2+/2-$ trials per schedule.

Table 14. Pretraining idealized exposure profile across the ten-schedule sweep at $F = 2$. Each schedule targets a nominal effective epoch count E, which at $F = 2$ corresponds to an idealized per-key exposure $E/F = E/2$. Both initialization regimes (CPT and from-scratch) share the same idealized profile.

Schedule group	E	E/F
(S1,E1) – (S4,E1)	1	0.5
(U,E4) / (P,E4)	4	2.0
(U,E8) / (P,E8)	8	4.0
(U,E16) / (P,E16)	16	8.0

E.1. Pretraining CPT Companion Figures

Figures 5 and 23 to 25 carry the four CPT keyed/null pairs across the (aligned, packed) surface and (exact, Gaussian) p-value-type combinations. Figures 26 and 27 carry the matching CPT watermark whole-model DIA AUC pairs, with aligned and packed surfaces shown side-by-side under each p-value type.

E.2. Pretraining From-Scratch Companion Figures

Figures 6 and 28 to 30 carry the four from-scratch keyed/null pairs across the (aligned, packed) surface and (exact, Gaussian) p-value-type combinations. Figures 31 and 32 carry the matching from-scratch watermark whole-model DIA AUC pairs, with aligned and packed surfaces shown side-by-side under each p-value type.

E.3. Pretraining Schedule Summaries

Tables 15 and 16 provide compact per-schedule summaries combining the realized $\hat{E}$ with the aligned and packed watermark $-\log_{10} p$ values for both initialization regimes.

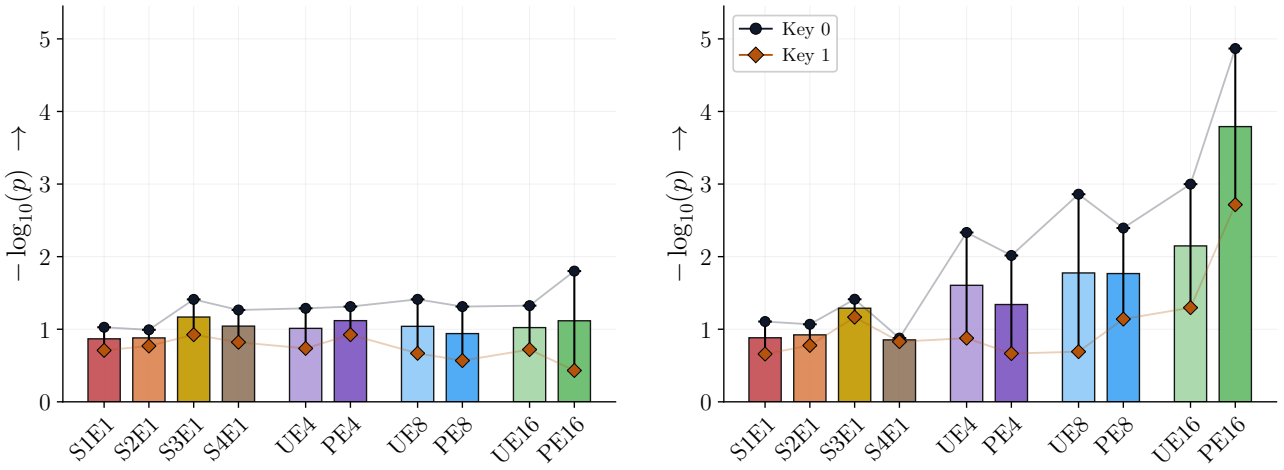

Figure 23. Pretraining CPT matched clean-twin false-probe null (left) and keyed signal (right) across the ten-schedule sweep at $F = 2$, on the aligned unpacked detection surface, scored as $-\log_{10} p$ under the empirical-Gaussian reference.

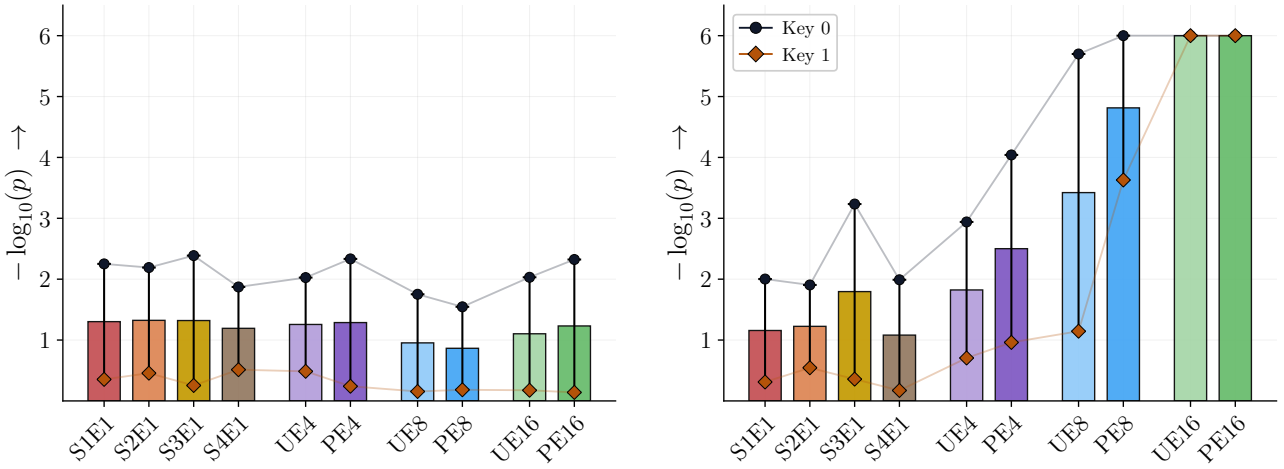

Figure 24. Pretraining CPT matched clean-twin false-probe null (left) and keyed signal (right) across the ten-schedule sweep at $F = 2$, on the packed detection surface, scored as $-\log_{10} p$ under the empirical-exact null. The packed surface is a more permissive oracle baseline than the aligned surface.

Table 15. CPT pretraining: schedule summary.

Schedule	Realized ($\hat{E}$)	Aligned WM ($-\log_{10}(p)$)	Packed WM ($-\log_{10}(p)$)
(S1,E1)	1.0000	0.8825	1.1630
(S2,E1)	1.0000	0.9225	1.2270
(S3,E1)	1.0000	1.2895	1.8080
(S4,E1)	1.0000	0.8535	1.0840
(U,E4)	3.8810	1.6050	1.8240
(P,E4)	4.0000	1.3405	2.5080
(U,E8)	7.8935	1.7755	3.4280
(P,E8)	8.0000	1.7670	5.7460
(U,E16)	16.2050	2.1480	12.9210
(P,E16)	16.0000	3.7915	33.5980

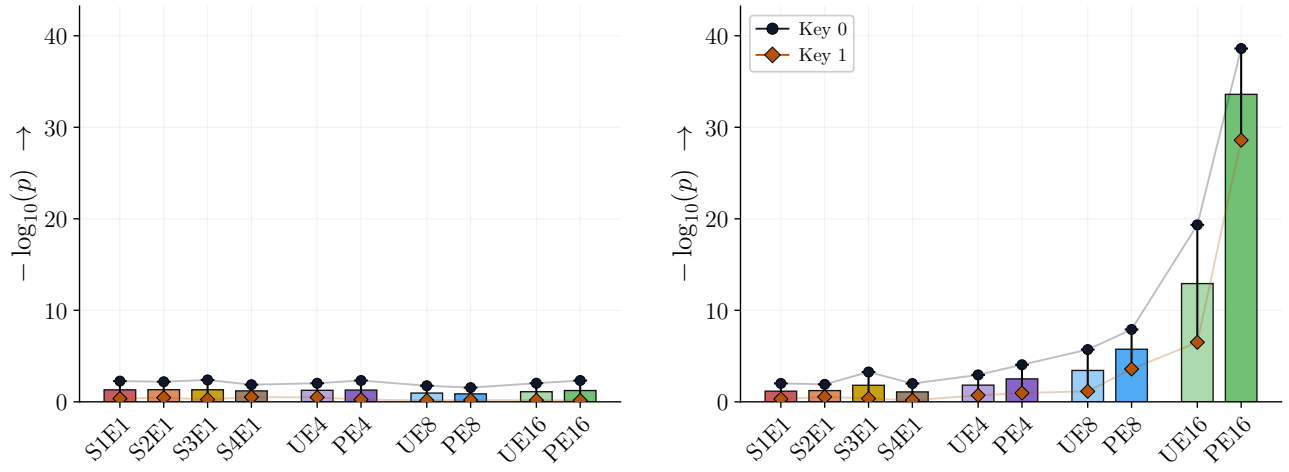

Figure 25. Pretraining CPT matched clean-twin false-probe null (left) and keyed signal (right) across the ten-schedule sweep at $F = 2$, on the packed detection surface, scored as $-\log_{10} p$ under the empirical-Gaussian reference.

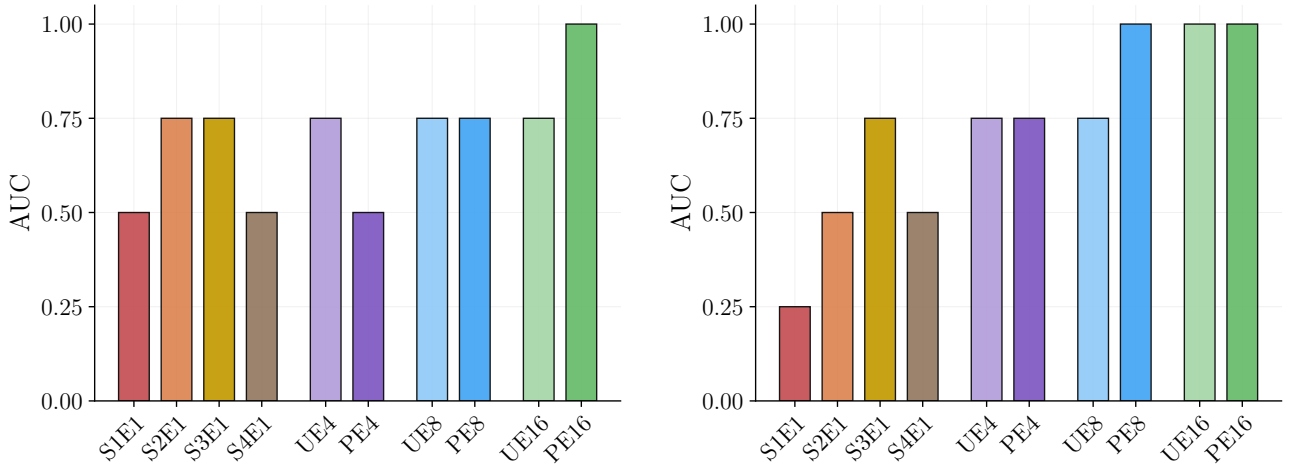

Figure 26. Pretraining CPT watermark whole-model DIA AUC across the ten-schedule sweep at $F = 2$, on the aligned (left) and packed (right) detection surfaces, both scored against an empirical-exact null. Each schedule contributes $2+/2-$ whole-model trials, so AUC is coarse but tracks the keyed-signal ordering of Figure 5.

Table 16. From-scratch pretraining: schedule summary.

Schedule	Realized ($\hat{E}$)	Aligned WM ($-\log_{10}(p)$)	Packed WM ($-\log_{10}(p)$)
(S1,E1)	1.0000	0.4025	0.8710
(S2,E1)	1.0000	0.6535	1.3460
(S3,E1)	1.0000	0.6415	2.7280
(S4,E1)	1.0000	0.8265	3.7550
(U,E4)	3.8895	1.5825	7.5010
(P,E4)	4.0000	0.9485	3.0750
(U,E8)	7.8115	3.9795	45.2420
(P,E8)	8.0000	3.2805	30.1350
(U,E16)	16.0740	48.4810	256.7420
(P,E16)	16.0000	31.4770	229.0900

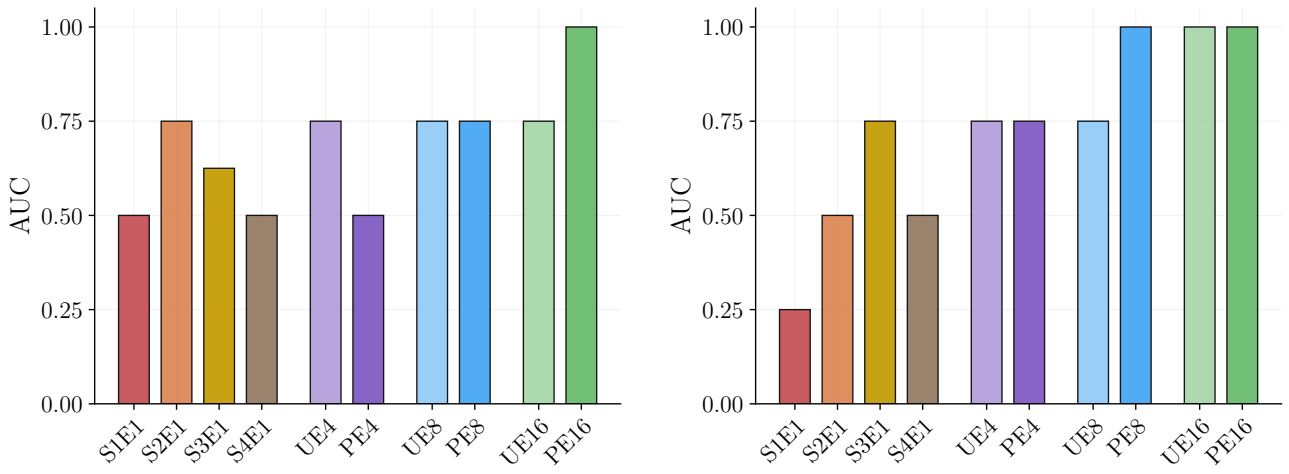

Figure 27. Pretraining CPT watermark whole-model DIA AUC across the ten-schedule sweep at $F = 2$, on the aligned (left) and packed (right) detection surfaces, both scored against an empirical-Gaussian null. Each schedule contributes $2+/2-$ whole-model trials.

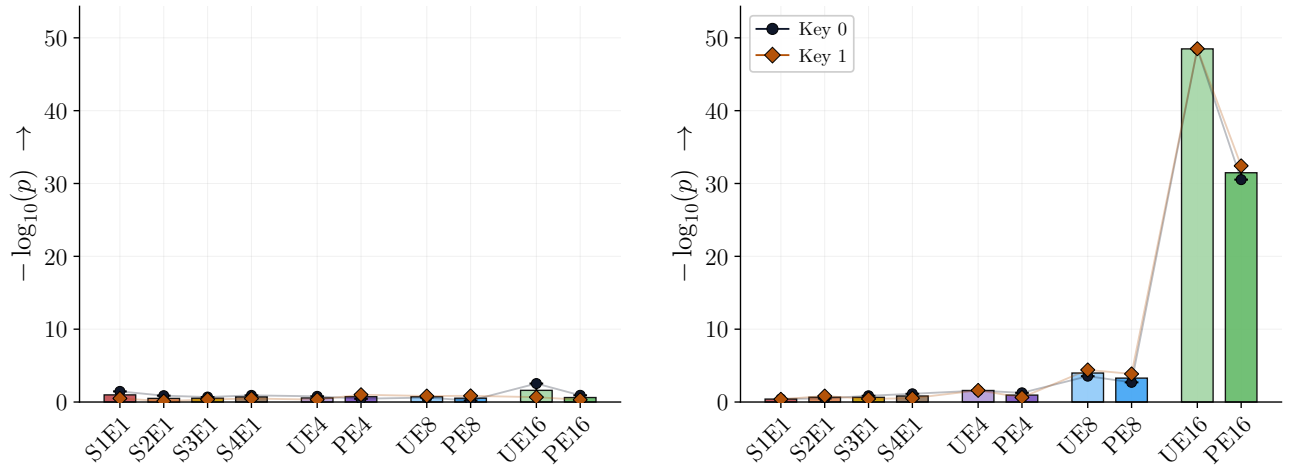

Figure 28. Pretraining from-scratch matched clean-twin false-probe null (left) and keyed signal (right) across the ten-schedule sweep at $F = 2$, on the aligned unpacked detection surface, scored as $-\log_{10} p$ under the empirical-Gaussian reference.

E.4. Pretraining Realized Exposure and $\hat{E}$ Per-Schedule Readbacks

Tables 17 and 18 report the realized normalized exposure $\hat{E}/F$ per schedule for both pretraining initialization regimes, the direct realized counterpart to the idealized E/F values in main-body Table 14. Tables 19 and 20 report the underlying realized $\hat{E}$ values. The corresponding idealized E targets are the integers in the schedule names (column E of Table 14).

E.5. Pretraining Training Scale and Trial Geometry

Tables 21 and 22 report the per-schedule watermarked-token totals (mean, min–max across paired models) and the corresponding fraction of each model's 10.49B-token training budget for the CPT and from-scratch initialization regimes respectively. Tables 23 and 24 report the per-schedule paired-model counts and the resulting n_+/n_- trial counts that drive each schedule's whole-model DIA AUC. Each schedule contributes $2/2$ positive/negative whole-model trials per init across both initialization regimes; Section 3.2 summarizes the extremal mass values.

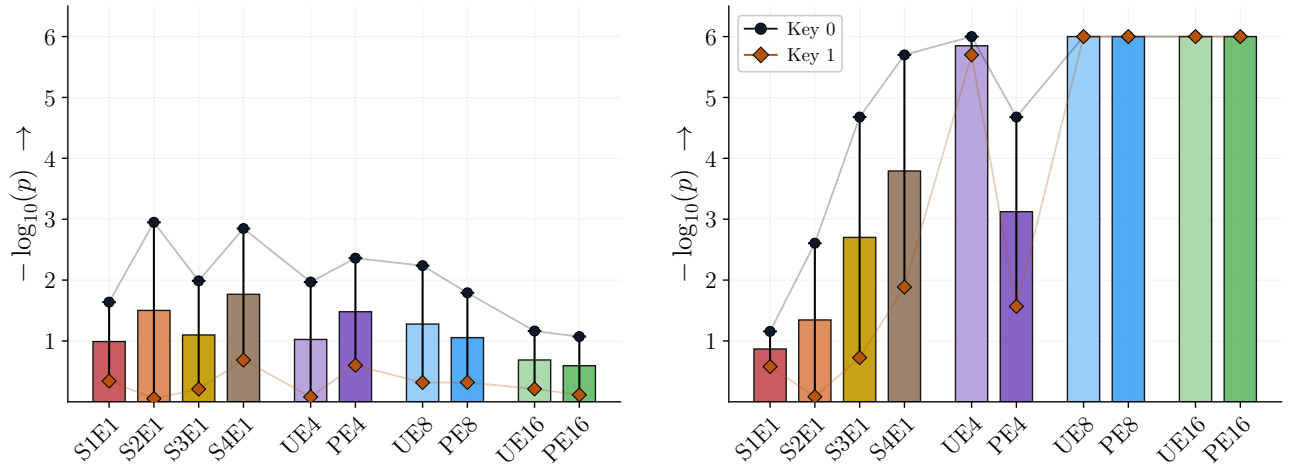

Figure 29. Pretraining from-scratch matched clean-twin false-probe null (left) and keyed signal (right) across the ten-schedule sweep at $F = 2$, on the packed detection surface, scored as $-\log_{10} p$ under the empirical-exact null. The packed surface is a more permissive oracle baseline than the aligned surface.

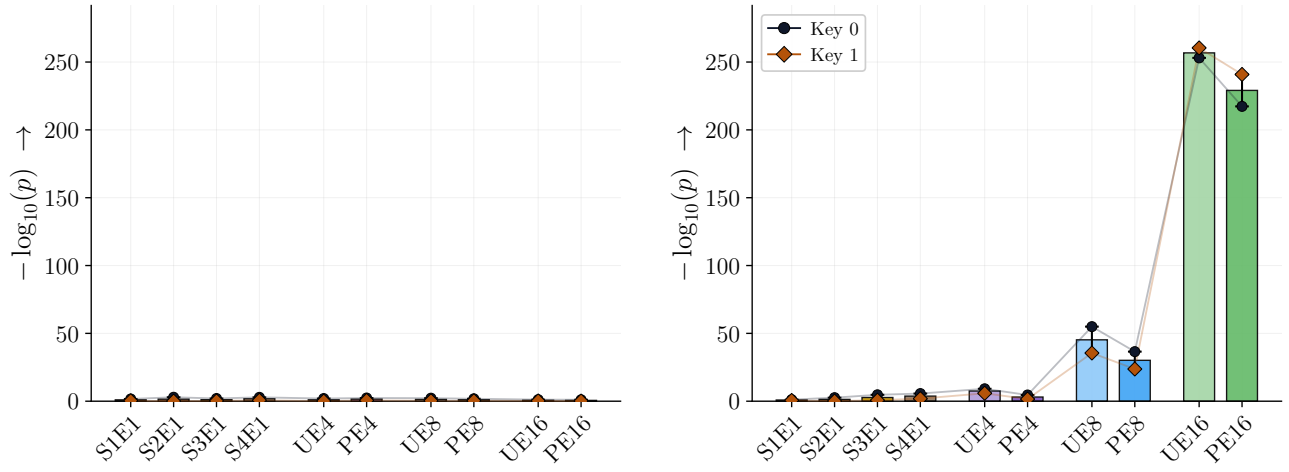

Figure 30. Pretraining from-scratch matched clean-twin false-probe null (left) and keyed signal (right) across the ten-schedule sweep at $F = 2$, on the packed detection surface, scored as $-\log_{10} p$ under the empirical-Gaussian reference.

Table 17. CPT pretraining: realized normalized exposure summary ($\hat{E}/F$).

Schedule	Realized ($\hat{E}/F$)
(S1,E1)	0.5000
(S2,E1)	0.5000
(S3,E1)	0.5000
(S4,E1)	0.5000
(U,E4)	1.9405
(P,E4)	2.0000
(U,E8)	3.9467
(P,E8)	4.0000
(U,E16)	8.1025
(P,E16)	8.0000

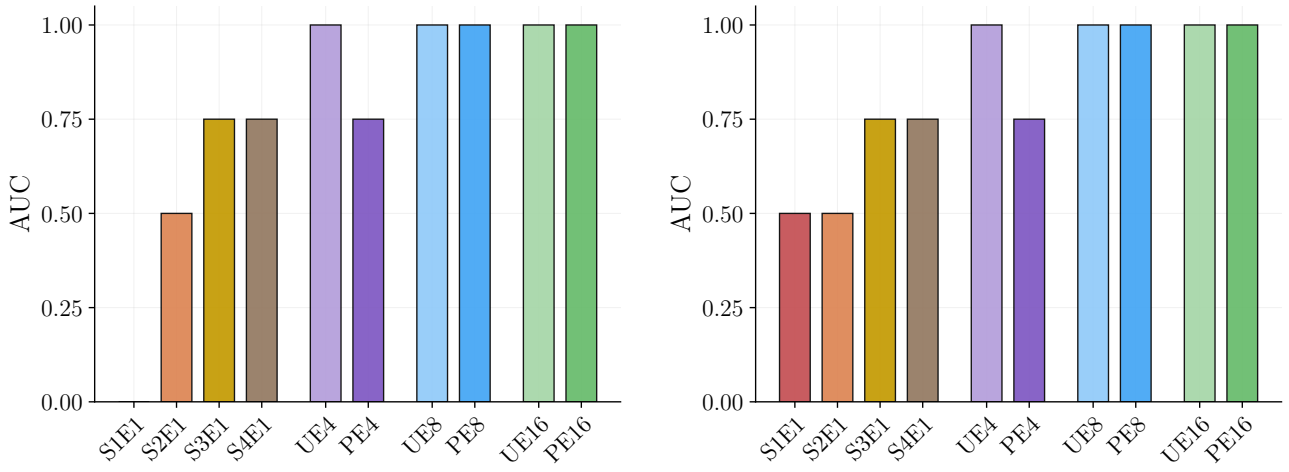

Figure 31. Pretraining from-scratch watermark whole-model DIA AUC across the ten-schedule sweep at $F = 2$, on the aligned (left) and packed (right) detection surfaces, both scored against an empirical-exact null. Each schedule contributes $2+/2-$ whole-model trials. The from-scratch DIA recovers near-saturated AUC at the high-exposure schedules well before CPT does.

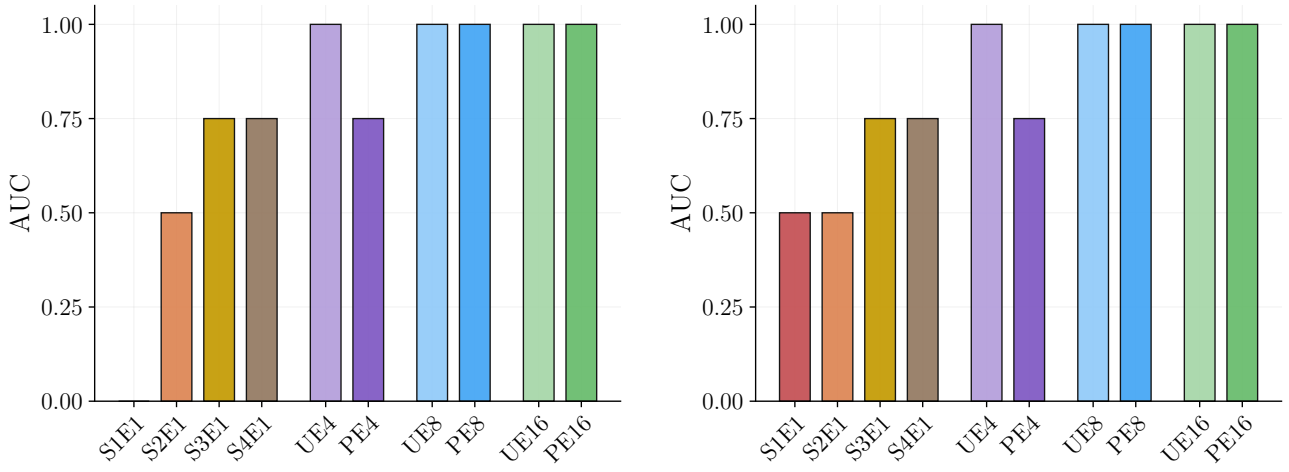

Figure 32. Pretraining from-scratch watermark whole-model DIA AUC across the ten-schedule sweep at $F = 2$, on the aligned (left) and packed (right) detection surfaces, both scored against an empirical-Gaussian null. Each schedule contributes $2+/2-$ whole-model trials.

Table 18. From-scratch pretraining: realized normalized exposure summary ($\hat{E}/F$).

Schedule	Realized ($\hat{E}/F$)
(S1,E1)	0.5000
(S2,E1)	0.5000
(S3,E1)	0.5000
(S4,E1)	0.5000
(U,E4)	1.9447
(P,E4)	2.0000
(U,E8)	3.9057
(P,E8)	4.0000
(U,E16)	8.0370
(P,E16)	8.0000

Table 19. CPT pretraining: realized exposure summary ($\hat{E}$).

Schedule	Realized ($\hat{E}$)
(S1,E1)	1.0000
(S2,E1)	1.0000
(S3,E1)	1.0000
(S4,E1)	1.0000
(U,E4)	3.8810
(P,E4)	4.0000
(U,E8)	7.8935
(P,E8)	8.0000
(U,E16)	16.2050
(P,E16)	16.0000

Table 20. From-scratch pretraining: realized exposure summary ($\hat{E}$).

Schedule	Realized ($\hat{E}$)
(S1,E1)	1.0000
(S2,E1)	1.0000
(S3,E1)	1.0000
(S4,E1)	1.0000
(U,E4)	3.8895
(P,E4)	4.0000
(U,E8)	7.8115
(P,E8)	8.0000
(U,E16)	16.0740
(P,E16)	16.0000

Table 21. CPT pretraining: training scale context. Per-schedule watermark token totals are relative to 10.49B train tokens per run. The percent columns report watermark-token share of total train tokens.

Schedule	WM tokens seen	Realized $\hat{E}$	Mean %	Range %
(S1,E1)	0.50M	1.000	0.005%	0.005%-0.005%
(S2,E1)	0.50M	1.000	0.005%	0.005%-0.005%
(S3,E1)	0.50M	1.000	0.005%	0.005%-0.005%
(S4,E1)	0.50M	1.000	0.005%	0.005%-0.005%
(U,E4)	1.94M-1.94M	3.877-3.885	0.019%	0.019%-0.019%
(P,E4)	2.00M	4.000	0.019%	0.019%-0.019%
(U,E8)	3.88M-4.01M	7.762-8.025	0.038%	0.037%-0.038%
(P,E8)	4.00M	8.000	0.038%	0.038%-0.038%
(U,E16)	8.10M-8.10M	16.197-16.213	0.077%	0.077%-0.077%
(P,E16)	8.00M	16.000	0.076%	0.076%-0.076%

E.6. Pretraining Loss-Based and Reference-Model Row-Level MIA and Whole-Model DIA Baselines

Tables 25 to 28 report the source-fold loss-based and reference-model row-level MIA and whole-model DIA AUCs alongside the watermark detector's own AUCs (which are the same values that drive Figures 26 and 31).

Table 22. From-scratch pretraining: training scale context. Per-schedule watermark token totals are relative to 10.49B train tokens per run. The percent columns report watermark-token share of total train tokens.

Schedule	WM tokens seen	Realized $\hat{E}$	Mean %	Range %
(S1,E1)	0.500M	1.000	0.005%	0.005%-0.005%
(S2,E1)	0.500M	1.000	0.005%	0.005%-0.005%
(S3,E1)	0.500M	1.000	0.005%	0.005%-0.005%
(S4,E1)	0.500M	1.000	0.005%	0.005%-0.005%
(U,E4)	1.938-1.950M	3.877-3.902	0.019%	0.018%-0.019%
(P,E4)	1.999M	4.000	0.019%	0.019%-0.019%
(U,E8)	3.827-3.982M	7.656-7.967	0.037%	0.036%-0.038%
(P,E8)	3.999M	8.000	0.038%	0.038%-0.038%
(U,E16)	7.973-8.096M	15.951-16.197	0.077%	0.076%-0.077%
(P,E16)	7.997M	16.000	0.076%	0.076%-0.076%

Table 23. CPT pretraining: model and watermark DIA trial geometry. In these pretraining cells, model counts and watermark DIA trial counts coincide because each target contributes one positive or negative whole-model trial to the pooled AUC.

Schedule	WM models n_+	Clean models n_-	WM DIA trials n_+/n_-
(S1,E1)	2	2	2 / 2
(S2,E1)	2	2	2 / 2
(S3,E1)	2	2	2 / 2
(S4,E1)	2	2	2 / 2
(U,E4)	2	2	2 / 2
(P,E4)	2	2	2 / 2
(U,E8)	2	2	2 / 2
(P,E8)	2	2	2 / 2
(U,E16)	2	2	2 / 2
(P,E16)	2	2	2 / 2

Table 24. From-scratch pretraining: model and watermark DIA trial geometry. In these pretraining cells, model counts and watermark DIA trial counts coincide because each target contributes one positive or negative whole-model trial to the pooled AUC.

Schedule	WM models n_+	Clean models n_-	WM DIA trials n_+/n_-
(S1,E1)	2	2	2 / 2
(S2,E1)	2	2	2 / 2
(S3,E1)	2	2	2 / 2
(S4,E1)	2	2	2 / 2
(U,E4)	2	2	2 / 2
(P,E4)	2	2	2 / 2
(U,E8)	2	2	2 / 2
(P,E8)	2	2	2 / 2
(U,E16)	2	2	2 / 2
(P,E16)	2	2	2 / 2

Table 25. CPT pretraining: row-level MIA AUC comparison.

	Watermark Readout	Loss-based Row MIA			
Schedule	$\mathrm{WM} - \log_{10}(p_{\mathrm{exact}})$	Raw-loss	Argmax	min-k_{10}	zlib
(S1,E1)	0.8810	0.5029	0.5021	0.5036	0.5023
(S2,E1)	0.9195	0.5165	0.5142	0.5206	0.5127
(S3,E1)	1.2870	0.5291	0.5247	0.5367	0.5225
(S4,E1)	0.8515	0.5079	0.5059	0.5104	0.5062
(U,E4)	1.6045	0.5497	0.5457	0.5612	0.5387
(P,E4)	1.3390	0.5765	0.5700	0.5939	0.5601
(U,E8)	1.7695	0.6104	0.6015	0.6333	0.5893
(P,E8)	1.7695	0.6610	0.6521	0.6932	0.6328
(U,E16)	2.1415	0.7197	0.7112	0.7540	0.6906
(P,E16)	3.7765	0.8163	0.8117	0.8636	0.7884

Table 26. CPT pretraining: fold-level whole-model DIA AUC comparison. Each cell contributes $2+/2-$ whole-model trials.

	Watermark DIA		Loss-based DIA			
Schedule	Aligned	Packed	Raw-loss	Argmax	min-k_{10}	zlib
(S1,E1)	0.5000	0.2500	0.7500	0.7500	0.7500	0.7500
(S2,E1)	0.7500	0.5000	1.0000	1.0000	1.0000	1.0000
(S3,E1)	0.7500	0.7500	1.0000	1.0000	1.0000	1.0000
(S4,E1)	0.5000	0.5000	1.0000	0.7500	1.0000	0.7500
(U,E4)	0.7500	0.7500	1.0000	1.0000	1.0000	1.0000
(P,E4)	0.5000	0.7500	1.0000	1.0000	1.0000	1.0000
(U,E8)	0.7500	0.7500	1.0000	1.0000	1.0000	1.0000
(P,E8)	0.7500	1.0000	1.0000	1.0000	1.0000	1.0000
(U,E16)	0.7500	1.0000	1.0000	1.0000	1.0000	1.0000
(P,E16)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Table 27. From-scratch pretraining: row-level MIA AUC comparison.

	Watermark Readout	Loss-based Row MIA			
Schedule	$\mathrm{WM} - \log_{10}(p_{\mathrm{exact}})$	Raw-loss	Argmax	min-k_{10}	zlib
(S1,E1)	0.4015	0.5028	0.5034	0.5064	0.5019
(S2,E1)	0.6520	0.5204	0.5130	0.5245	0.5141
(S3,E1)	0.6395	0.5823	0.5648	0.6058	0.5628
(S4,E1)	0.8250	0.6025	0.5957	0.6517	0.5797
(U,E4)	1.5800	0.6846	0.6621	0.7379	0.6525
(P,E4)	0.9470	0.6429	0.6043	0.6794	0.6106
(U,E8)	4.0410	0.8418	0.8209	0.8848	0.8122
(P,E8)	3.2730	0.8472	0.8072	0.8937	0.8131
(U,E16)	6.0000	0.9744	0.9768	0.9602	0.9715
(P,E16)	6.0000	0.9792	0.9764	0.9711	0.9765

Table 28. From-scratch pretraining: fold-level whole-model DIA AUC comparison. Each cell contributes $2+/2-$ whole-model trials.

	Watermark DIA		Loss-based DIA			
Schedule	Aligned	Packed	Raw-loss	Argmax	min-k_{10}	zlib
(S1,E1)	0.0000	0.5000	0.7500	1.0000	1.0000	0.5000
(S2,E1)	0.5000	0.5000	1.0000	1.0000	1.0000	1.0000
(S3,E1)	0.7500	0.7500	1.0000	1.0000	1.0000	1.0000
(S4,E1)	0.7500	0.7500	1.0000	1.0000	1.0000	1.0000
(U,E4)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(P,E4)	0.7500	0.7500	1.0000	1.0000	1.0000	1.0000
(U,E8)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(P,E8)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(U,E16)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(P,E16)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000