WATERMARK BASINS: BLACK-BOX DETECTION OF DISTILLED LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Proprietary language models and their reasoning traces are vulnerable to API-based distillation, where an adversary trains a student model on the teacher model outputs. To provide evidence of distillation, model providers can watermark the teacher outputs, aiming to leave radioactivity in the student outputs. Existing black-box detectors search for such radioactivity using token-level statistics from the suspect model's outputs. However, token-level signals are not always detectable; for instance, when the teacher model is distilled at a low temperature. In this work, we discover *watermark basin*, a persistent signal that enables more reliable watermark detection: during fine-tuning, the watermarked teacher traces embed a distinct, watermark key-specific *geometric signal in the parameter space* that lies orthogonal to the ordinary task-specific fine-tuning subspace. Building on this, we introduce Watermark Basin Alignment (WM-Basin), a black-box watermark detector that queries through the suspect model's API, without requiring access to its weights or logits. In particular, WM-Basin maps the generated text into gradient representations using a fixed clean probe model, projects out the ordinary fine-tuning task directions, and verifies if the residual gradient is aligned with the watermark basin induced by the watermark key. We evaluate WM-Basin extensively across diverse teacher models and student architectures (spanning 135M to 8.2B parameters) distilled on mathematical reasoning. Under a strict 0.1% false-positive rate, WM-Basin detects 79% of distilled models at n-gram=1 and 45% at n-gram=3, substantially improving the state-of-the-art token-level baseline (29% and 1.3%, respectively). Furthermore, because the watermark signal resides in a persistent basin, it survives aggressive post-distillation modifications, including further fine-tuning, parameter quantization, pruning, and reinforcement-learning post-training.

1 INTRODUCTION

Proprietary language models are increasingly targeted by API-based distillation: an adversary queries a closed teacher model and fine-tunes a competing student model on its outputs. This attack extends black-box model extraction, first demonstrated against ML-as-a-service systems (Tramèr et al., 2016), and has become a major threat to reasoning LLMs (Anthropic, 2026a; Google Threat Intelligence Group, 2026; Remler & Hayum, 2026). Providers have responded by withholding reasoning traces, but concealment does not prevent distillation. Hidden traces can be extracted at scale from encrypted response blocks (Panfilov et al., 2026), while answers and short summaries can be expanded into synthetic traces that recover much of the benefit of training on

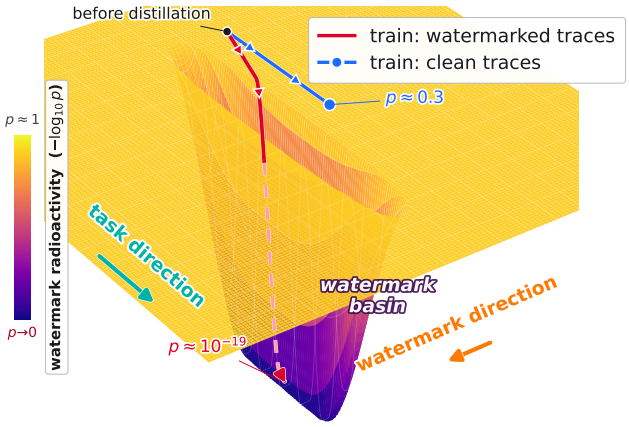


Figure 1: Training on watermarked traces nudges the student into a persistent, key-specific **watermark basin** that survives post-training. Our paper introduces Watermark Basin Alignment (WM-Basin), a method to recover this geometry solely from black-box outputs for distillation detection.

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.

the originals (Zhang et al., 2026). Other defenses make extraction harder rather than detectable, by monitoring extraction-like querying (Anthropic, 2026a; Google Threat Intelligence Group, 2026) or degrading what the API returns (Savani et al., 2026; Ma et al., 2026); each costs benign users utility or assumes the adversary stays observable while collecting (Remler & Hayum, 2026). Classical model watermarking embeds a detectable signal in the owner's own released model (Uchida et al., 2017; Adi et al., 2018), whereas the suspect here is the student model that the owner never trained. App. A gives the full discussion on related work.

In this work, we focus on output watermarking, which is encouraged by emerging regulatory requirements (European Commission, 2026). Anthropic, for example, embeds watermarks in texts generated by Claude models released beginning in August 2026 (Anthropic, 2026b). Such watermarks can survive distillation: Sander et al. (2024) show that students trained on watermarked teacher outputs retain a "radioactive" signal. Their black-box detector, to our knowledge the only existing one for this setting and our baseline throughout, tests suspect outputs for an excess of watermark-favored tokens, but only where watermark-conditioned contexts recur. We find that this dependence is fragile: detection degrades sharply with larger context windows and loses most of its power under low-temperature teacher sampling. *We therefore seek a watermark signal that does not require exact token-level recurrence.*

Our major finding is the **Watermark Basin**: supervised fine-tuning on clean (watermark-free) teacher traces moves a student along a broad *task direction*, whereas watermarked traces add a faint, key-specific *watermark direction* (Fig. 1), leading the student into a key-specific region. Post-training updates cannot easily push the student out of this basin. Building on this geometry, we introduce **Watermark Basin Alignment (WM-Basin)**, a black-box detector that represents the suspect model outputs and the watermarked teacher model outputs as gradients, using one fixed clean probe model. The intuition is that after removing the task subspace estimated from the clean model, the distilled student should align with the watermark key *more closely* than other decoy keys (which were not used for watermarking the teacher model outputs). WM-Basin turns this relative rank information into a calibrated p value using a non-parametric ranking test (Dwass, 1957), i.e., a small value provides the evidence of distillation from the claimed teacher and watermark key on the audited domain (Fig. 2).

We evaluate WM-Basin broadly along many axes: the student model architecture, the teacher model queried, the watermarking scheme, the sampling temperature and the watermarked fraction of the distillation traces, the fine-tuning recipe, and the task domain. Among $1{,}200$ student models distilled from the DeepSeek-R1-Distill-Qwen-14B model (Guo et al., 2025) on OpenR1-Math reasoning traces (Hugging Face, 2025), at a 0.1% false-positive rate, WM-Basin detects 79% of distillation at n-gram$=1$ and 45% at n-gram$=3$, against 29% and 1.3% for the state-of-the-art black-box detector (Sander et al., 2024). In addition, we show that quantization, pruning, and reinforcement-learning post-training all leave our detection results intact, as none of them moves the student model out of the watermark basin. Like any black-box method, WM-Basin reads only the suspect model's text, so detection depends on that text carrying the teacher's token distribution. Continued training on clean data can reshape that distribution and therefore degrade detection performance, although the basin itself survives in the weights and enables identification easily under white-box access, which may be of independent interest in other distillation detection scenarios. Our contributions are as follows:

- **Discovery of Watermark Basins:** We discover watermarked distillation embeds a persistent basin in all distilled students formed by the ordinary task direction and a watermark key-specific, faint watermark direction. It is difficult to leave this basin without knowledge of the key: post-training operations (quantization, pruning, supervised fine-tuning, RL) displace weights primarily along the task axis, keeping the key-specific information intact, which we exploit next for detection.
- **Parameter-space black-box detector (WM-Basin):** We propose WM-Basin, which looks for distillation evidence in the parameter space. By mapping text outputs to gradients of a fixed clean probe model, removing the shared task subspace and projecting back into parameter space, and scoring residual alignment against decoy keys, the watermark becomes far more detectable than in token space, with exact finite-sample FPR control and no suspect parameter access.
- **State-of-the-art robust detection at scale:** Across 24 student architectures (135M to 8.2B), 4 teachers, and 4 watermarking schemes, WM-Basin achieves 79% TPR at 0.1% FPR vs. 29% for the state-of-the-art token-level detector, and remains effective under quantization, pruning, and RL while preserving student utility.

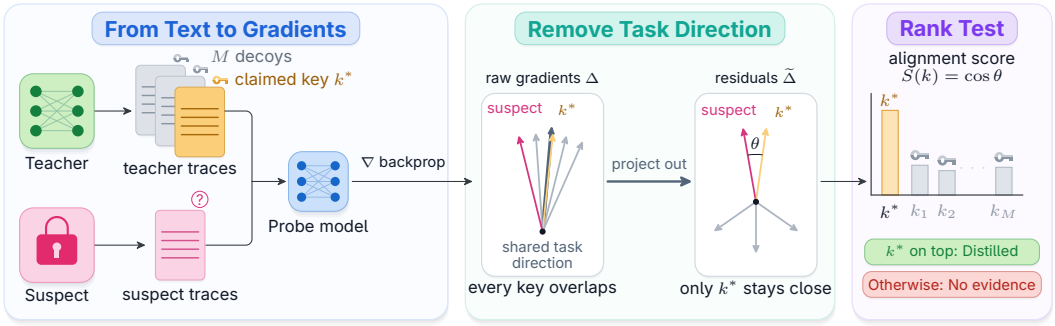


Figure 2: **WM-Basin: black-box distillation detection via watermark basin alignment.** The three panels are the three stages of §2.3. **(i)** The verifier collects suspect outputs on test prompts and, on separate build prompts, teacher outputs under the claimed watermark key and M decoy keys. Backpropagating each collection through one fixed probe model gives the suspect gradient Δ_{sus} and one reference gradient Δ_k per key. **(ii)** Raw gradients share large common task components that inflate the cosine similarities, i.e., every key overlaps the suspect. Projecting out the shared task directions leaves the residuals $\widetilde{\Delta}_{\text{sus}}$ and $\widetilde{\Delta}_k$, among which only the claimed key stays closer to the suspect; θ is the angle between them. **(iii)** Each candidate key receives the alignment score $S(k) = \cos(\widetilde{\Delta}_{\text{sus}}, \widetilde{\Delta}_k)$, the cosine of that angle, and the claimed key's rank among the decoys gives p_{rank}. The claimed key ranking at the top is the evidence of distillation; a decoy key at the top is not.

2 Watermark-Basin Detection

2.1 Threat Model and Statistical Setup

Distillation setting and adversary model. A proprietary teacher model $\mathcal{M}_{\text{teacher}}$ serves completions watermarked under a secret key $k^\star$. $k^\star \leftarrow \mathsf{KeyGen}$ is generated from the watermark scheme's key-generation procedure. Following deployed systems (Aaronson, 2022; Dathathri et al., 2024; Sander et al., 2026; Anthropic, 2026b), we consider distortion-free schemes: the teacher's marginal next-token distribution is preserved, but for each prompt x, the output $y \sim \mathcal{M}_{\text{teacher}}(x; k^\star)$ is coupled to the pseudorandomness induced by $k^\star$. The adversary adaptively chooses N prompts $\mathcal{X}_{\text{train}} = \{x_i\}_{i=1}^N$, receives watermarked traces $\{y_i\}$, and builds $\mathcal{D} = \{(x_i, y_i)\}$. The adversary trains $\mathcal{M}_{\text{sus}}$ via supervised fine-tuning on $\mathcal{D}$. Without knowledge of $k^\star$, watermarked completions are indistinguishable from clean text, so the default adversary is unaware of watermarking and uses traces as-is. After fine-tuning on $\mathcal{D}$, the adversary may further post-train $\mathcal{M}_{\text{sus}}$ on clean data, e.g., via continued SFT or GRPO (Guo et al., 2025).

Verifier regime. An audit is opened by the verifier, the model owner or some authorized third party. The audit begins by naming one capability the owner sells and the suspect appears to compete on. We call it the *audited domain* D (e.g., math) and let P_D be its induced prompt distribution; the verifier chooses both and draws every prompt it uses from P_D. It holds two disjoint pools: a *build* pool used to construct the detector and a disjoint *test* pool used to read the suspect. Neither overlaps the adversary's prompts, which the verifier never sees; separating build from test prompts avoids score inflation from prompt reuse. The verifier's access to $\mathcal{M}_{\text{sus}}$ is *black-box*, observing only texts sampled on the test prompts. The access is also *unsupervised* in Sander et al. (2024)'s sense: it does not hold the traces $\mathcal{M}_{\text{sus}}$ was trained on, only fresh teacher traces on the build prompts, unwatermarked ones and watermarked ones under key $k^\star$ (and also under M independent decoy keys $k_j \leftarrow \mathsf{KeyGen}$), together with a pool of public clean checkpoints. It knows the system prompts used to query $\mathcal{M}_{\text{sus}}$ and the teacher, because it chooses them. The number of test prompts to the suspect is the query budget.

Statistical objective. Let $S(k)$ be any scalar alignment score computed from the suspect's completions on held-out test prompts and from reference teacher traces on separate build prompts under key k, as instantiated in §2.3. Its absolute magnitude is not comparable across suspects, prompt pools, or probe models, so an audit reads only where the claimed key $k^\star$ ranks among decoy keys. Our null hypothesis H_0 is that the suspect's outputs on the audited domain D contain no information about the production key $k^\star$, where $k^\star \leftarrow \mathsf{KeyGen}$. In particular, if the texts do depend on $k^\star$, the

dependence can only arise via the distillation path $k^\star \to \{y_i\} \to \theta_{\text{sus}} \to \mathcal{M}_{\text{sus}}$, so rejecting H_0 establishes *watermark radioactivity*: the suspect was trained on the teacher's keyed text, which separates it from any model not distilled under $k^\star$. A failure to reject is scoped to domain D and we measure the detection power beyond the distillation domain in §3.2.

2.2 Watermark Basins in Parameter Space

We first use a white-box setting to *investigate* what distinguishes a student trained on watermarked traces from one trained on clean traces, and thus what makes a model radioactive. Fig. 1 presented the watermark basin; here we test that geometry directly with access to the student weights. We fix the true key $k^\star$ and fine-tune the same base model on either teacher traces watermarked with $k^\star$, giving $\boldsymbol{\theta}_{\text{student}}$, or unwatermarked traces, giving a clean control $\boldsymbol{\theta}_{\text{ctrl}}$. The teacher, prompts, initialization, and training recipe are otherwise identical, so both students learn the same task while only one sees the watermark. Let $\Delta_{\text{total}} := \boldsymbol{\theta}_{\text{student}} - \boldsymbol{\theta}_{\text{base}}$ be the total update from watermarked training and $\Delta_{\text{task}} := \boldsymbol{\theta}_{\text{ctrl}} - \boldsymbol{\theta}_{\text{base}}$ the clean task update. We subtract from Δ_{total} the component parallel to Δ_{task}. The residual Δ_{wm} captures the watermark-specific part of the update:

$$\Delta_{\text{wm}} := \Delta_{\text{total}} - \frac{\langle \Delta_{\text{total}}, \Delta_{\text{task}} \rangle}{\|\Delta_{\text{task}}\|_2^2} \Delta_{\text{task}}. \tag{1}$$

We normalize Δ_{wm} and Δ_{task} to obtain the watermark axis $\hat{\Delta}_{\text{wm}}$ and task axis $\hat{\Delta}_{\text{task}}$, respectively.

To visualize the basin, we start at $\boldsymbol{\theta}_{\text{student}}$ and sweep the weights separately along $\hat{\Delta}_{\text{wm}}$ and $\hat{\Delta}_{\text{task}}$. At each point, we apply the open-weight detector (Sander et al., 2024) to the perturbed student model and plot the resulting $-\log_{10} p$, where a smaller p-value provides stronger evidence of distillation. From Fig. 1 we can see that the surface is strongly anisotropic: a small step along the watermark axis $\hat{\Delta}_{\text{wm}}$ causes the watermark signal to collapse, whereas much larger displacements along the task axis $\hat{\Delta}_{\text{task}}$ leave it detectable, revealing broad tolerance to task-directed change but a sharp boundary along $\hat{\Delta}_{\text{wm}}$ (Figure 4 gives a more detailed presentation of the basin).

This asymmetry gives the two axes different roles. Its width along $\hat{\Delta}_{\text{task}}$ suggests why ordinary post-training can alter task behavior without immediately erasing the basin, a hypothesis tested under compression and continued training in §3.3. Its narrowness along $\hat{\Delta}_{\text{wm}}$ makes the basin key-specific, motivating comparison against decoy keys rather than thresholding a generic watermark score. This analysis relies on access to the student's parameters, which a black-box verifier does not have. In the next section, we show how to detect the same key-specific signal using only text sampled from the suspect.

2.3 WM-Basin: Black-Box Estimation via Probe Gradients

A black-box verifier cannot inspect the suspect's parameters and observes only its generated text. WM-Basin maps suspect, keyed-teacher, and clean-model outputs into a shared gradient representation, removes variation common to clean models, and ranks the claimed key by the remaining alignment. Fig. 2 gives the overview, and Algorithm 1 in App. B.2 gives the operational split.

Stage (i): from text to probe gradients. We use a small clean model as the probe and warm its LoRA parameters θ_{probe} on held-out unwatermarked teacher traces. We then freeze θ_{probe}. Throughout this section, ∇ denotes differentiation with respect to these fixed probe parameters, so a prompt–completion pair (x, y) is represented by $\nabla \log p_{\theta_{\text{probe}}}(y \mid x)$. This gradient is the parameter nudge that would make the probe more likely to produce y given x. If the suspect has inherited watermark key k, the nudges induced by its outputs should contain the same key-specific component as those induced by teacher outputs under k; this alignment is the signal we seek to measure. We average these vectors over suspect outputs on the test prompts to obtain Δ_{sus}, over teacher outputs on separate build prompts to obtain one reference gradient Δ_k per key, and over clean-model outputs to obtain the gradients used in stage (ii). These gradients provide a common coordinate system for comparing first-order fine-tuning tendencies; they do not reconstruct the suspect's hidden weight update (Xia et al., 2024). A single fixed random projection then maps every gradient to d=8,192 dimensions while approximately preserving inner products Johnson et al. (1984). This stage needs only keyed teacher traces, not the watermark scheme's own detector.

Stage (ii): removing the shared task subspace. Raw probe gradients share large components caused by the task, response format, and generic language modeling. These common components

inflate every cosine similarity $\cos(\Delta_{\text{sus}}, \Delta_k)$, obscuring the smaller differences that distinguish one key from another. We therefore adapt the *All-but-the-Top* post-processing method for dehubbing learned representations (Mu et al., 2017): clean checkpoints generate responses on held-out build prompts, and PCA on their probe gradients identifies the top q directions of clean-model variation. We collect these orthonormal directions in $U = [u_1, \ldots, u_q] \in \mathbb{R}^{d \times q}$. For every raw gradient Δ, we explicitly subtract its projection onto those top directions, $\widetilde{\Delta} = (I - UU^\top)\Delta$. Next, we compute the score

$$S(k) = \cos\big(\widetilde{\Delta}_{\text{sus}}, \widetilde{\Delta}_k\big). \tag{2}$$

We obtain U from clean models alone, never on suspects, and choose q=100. More details can be found in App. C.5.

Stage (iii): ranking against decoy keys. Because score S has no absolute scale, the verifier evaluates it at the claimed key and at M independent decoy keys $k_1, \ldots, k_M \leftarrow \mathsf{KeyGen}$. After that, the verifier reads the claimed key's rank:

$$p_{\text{rank}} = \frac{1 + \sum_{j=1}^{M} \mathbb{I}(S(k_j) \geq S(k^\star))}{M + 1}. \tag{3}$$

Under H_0, the claimed key and decoys are independent outputs of the same key generator and pass through the same probe, random projection, basis U, and prompts. Their scores are therefore exchangeable, giving p_{rank} exact finite-sample validity marginally over the draw of $k^\star$ (Dwass, 1957; Pesarin & Salmaso, 2010). Intuitively, p_{rank} should be uniformly distributed from 0 to 1, under H_0; conversely, if p_{rank} is very small, then the verifier claims distillation from $k^\star$ with high confidence. The minimum attainable value is $1/(M+1)$, requiring $M \geq 99$ or 999 to certify $\alpha = 0.01$ or 0.001.

The reference bank $\{\Delta_k\}$ and U are built once and reused: each fresh audit (on a new suspect model) requires only fresh suspect queries, results of Δ_{sus} and $\widetilde{\Delta}_{\text{sus}}$, and the final scoring and ranking. The computational cost is dominated by the one-time generation of teacher traces under $M + 1$ keys.

3 Experiments and Results

In this section, we present the experimental results, focusing primarily on n-gram=3. Full results across all evaluated n-gram windows are provided in App. D.

3.1 Experimental Setup

Distillation. We evaluate distillation detection across 24 suspect model architectures, 4 teacher models, and 4 watermarking schemes. In the default setting, the teacher is DeepSeek-R1-Distill-Qwen-14B watermarked with TextSeal (Guo et al., 2025; Sander et al., 2026), a dual-key scheme that samples each token under one of two secret keys, mixed at a fixed rate α=0.1. Following standard reasoning distillation protocols, an adversary fine-tunes a Qwen2.5-3B base model (Yang et al., 2024) for math on 1,900 quality-filtered OpenR1-Math-220k traces (Hugging Face, 2025) from that teacher, all generated under one fixed math-teacher system prompt, which the suspect model is fine-tuned conditioned on. We evaluate capability on held-out GSM8K (Cobbe et al., 2021) and Math-500 (Hendrycks et al., 2021), and defer the full distillation setup to App. C.2.

Sampling suspect model outputs. The verifier chooses one domain to audit and draws every prompt it uses for detection from that domain; it never sees the adversary's prompts or traces (§2.1). By default the audited domain is OpenR1-Math-220k (Hugging Face, 2025), held out from the adversary's distillation prompts, and every audit stage uses the same system prompt as the distillation traces, a *matched-prompt* audit; §3.2 varies both the domain and the system prompt. We read each suspect on 1,900 prompts from the test pool $\mathcal{X}_{\text{test}}$ (disjoint from the distillation prompts $\mathcal{X}_{\text{train}}$) and score the first 512 tokens of every response, around 1M tokens per suspect.

Detector. WM-Basin turns text into a gradient with a probe model, by default a clean Qwen2.5-3B base warmed on around 450 clean teacher traces from the audited domain (App. D.6 measures both the choice of probe model and the warm-up). On 512 build-pool prompts we query the teacher under the claimed watermark key and 999 decoys, then map those responses and the suspect's generations through the probe, giving a 1,000-key reference gradient bank and the suspect gradient scored against it. Before scoring we project both away from a PCA task subspace fit on 43 off-the-shelf checkpoints,

Table 1: **Distillation detection across 24 student architectures.** A single TextSeal-watermarked teacher (DeepSeek-R1-Distill-Qwen-14B) is distilled into 24 students. Each cell is the percentage of up to 50 distilled models per architecture flagged at a p-value of 5%, 1%, or 0.1%; **Overall** pools all models in that column (1,200 for each), and `Acc.` is post-distillation GSM8K accuracy.

Student structure	n-gram=1							n-gram=2							n-gram=3						
	Acc.	WM-Basin			Black-box			Acc.	WM-Basin			Black-box			Acc.	WM-Basin			Black-box		
		TPR@FPR			TPR@FPR				TPR@FPR			TPR@FPR				TPR@FPR			TPR@FPR		
		5%	1%	.1%	5%	1%	.1%		5%	1%	.1%	5%	1%	.1%		5%	1%	.1%	5%	1%	.1%
Qwen3-4B-Instruct	88.7	100	100	100	63	26	9	89.5	100	100	100	38	18	8	91.6	100	100	88	8	2	0
Qwen3-4B-Instruct-2507	91.4	100	100	100	56	33	14	91.6	100	100	100	36	22	8	90.8	98	96	90	14	4	0
Qwen3-8B-Instruct	89.6	100	100	100	49	19	2	90.8	100	100	100	20	12	0	89.9	100	100	92	6	0	0
Qwen2.5-7B-Instruct	87.8	100	100	100	70	47	12	87.9	100	100	98	42	24	8	88.8	100	92	80	18	2	0
Qwen2.5-Math-7B-Instruct	83.1	100	100	100	42	23	2	85.6	100	100	96	34	10	2	88.0	100	100	76	12	4	0
DeepSeek-R1-Distill-Qwen-7B	86.7	100	100	100	44	7	0	87.4	100	100	98	24	4	2	87.0	100	96	68	0	0	0
Qwen3-8B-Base	88.6	95	91	74	51	28	5	89.1	80	70	46	40	20	8	87.0	62	48	24	10	6	0
Llama-3.1-8B-Instruct	84.1	100	100	100	91	77	42	84.5	100	100	88	80	52	18	84.4	98	80	66	24	4	4
Qwen3-1.7B-Instruct	77.6	100	100	100	42	19	5	80.1	100	100	96	28	16	2	81.7	98	88	70	4	0	0
Qwen2.5-3B-Instruct	77.7	100	100	100	72	56	19	77.1	100	98	94	50	26	8	79.7	100	90	58	18	6	0
DeepSeek-R1-Distill-Qwen-1.5B	75.7	100	100	100	33	2	2	77.3	100	100	94	12	2	0	74.8	100	94	76	4	0	0
Llama-Guard-3-8B	61.7	100	100	100	100	84	51	62.6	100	96	76	82	64	22	63.0	92	70	52	24	6	2
Qwen3-0.6B-Instruct	61.3	100	100	100	72	42	16	62.3	100	98	88	30	16	2	59.6	96	74	50	10	0	0
Qwen2.5-1.5B-Base	55.9	100	100	100	74	35	7	52.2	98	90	66	38	20	6	58.5	82	62	38	12	6	0
Qwen2.5-1.5B-Instruct	58.2	100	100	98	74	44	16	58.5	92	80	54	42	10	0	54.6	68	52	34	8	2	0
Llama-3.2-1B-Instruct	36.0	100	100	98	70	42	16	35.8	98	92	72	52	30	16	38.7	96	76	44	14	6	0
Qwen2.5-0.5B-Base	31.9	100	100	91	88	65	44	30.4	92	78	56	66	44	12	36.8	88	62	20	16	6	0
Qwen2.5-0.5B-Instruct	30.7	100	100	95	79	54	19	31.6	94	84	58	54	28	4	30.4	92	64	26	18	4	2
SmolLM2-1.7B-Base	18.5	95	86	56	26	5	2	18.0	82	58	34	6	0	0	18.0	64	34	12	4	0	0
Gemma-2-2B-Base	10.8	47	19	7	100	98	88	10.4	12	6	2	86	72	44	5.9	14	4	0	30	14	0
Llama-Guard-3-1B	3.8	100	84	56	86	54	21	4.4	88	64	38	46	24	10	5.8	80	42	16	16	6	2
Gemma-3-1B-Base	1.8	56	26	2	84	44	14	1.7	38	14	4	46	20	0	1.8	32	14	4	16	4	0
SmolLM2-135M-Instruct	1.5	56	28	7	7	5	0	1.6	50	18	10	6	4	0	1.8	34	16	2	4	0	0
SmolLM2-135M-Base	1.1	56	26	2	12	5	0	1.4	48	18	6	6	2	0	1.3	32	12	4	0	0	0
Overall		**92**	**86**	**79**	**71**	**52**	**29**		**86**	**78**	**66**	**52**	**29**	**12**		**80**	**65**	**45**	**20**	**7**	**1**

which leaves the key-specific alignment the rank test reads (§2.3). All three stages draw separately from the build pool $\mathcal{X}_{\text{build}}$, disjoint from $\mathcal{X}_{\text{test}}$. Ranking the claimed key gives p-values down to 0.001; unless stated otherwise we report true positive rate (TPR) at $p \leq 0.01$ (1% false positive rate). We validated false positive rate in App. C.6.

Baseline. We compare with the black-box sampled-token detector of Sander et al. (2024), to our knowledge the only existing detector for this setting; it reads the same suspect text as WM-Basin, so every comparison is paired. We run it under its own protocol with both filters that give it power: a position is scored only when its context window occurs in a held-out corpus of 5,000 teacher traces regenerated under the key being scored, and only on distinct (context, token) pairs. Its p-value is computed analytically from the score's null distribution rather than by ranking decoys, and we threshold both detectors at the same p-value. App. C.4 gives the test used for each watermark scheme and lists the remaining settings.

3.2 Detection performance

Effect of the student architecture. Does detection depend on which student architecture the adversary selects? We vary the student across 24 architectures spanning 6 families (135M to 8.2B parameters), distilling 50 models for each student architecture, resulting in 1,200 distilled models for n gram. We hold the teacher model, distillation recipe, probe model, and reference gradient bank fixed. WM-Basin outperforms the existing baseline, and its margin over the baseline widens with the n-gram window. At n-gram=3 and $p \leq 0.001$, the baseline recovers almost nothing while WM-Basin detects 45% of distilled models. Furthermore, detectability is governed by task acquisition rather than architecture: WM-Basin mostly misses student models that failed to learn the task, indicating that evading detection means giving up that capability.

Effect of the distillation temperature. What happens when the adversary samples its distillation traces at a temperature the verifier does not know? We generate the teacher's traces at T=1.0, 0.7, and 0.5 and distill 100 student models from each setting. A lower T concentrates the teacher's

Table 2: **Distillation detection across distillation temperatures.** Qwen2.5-3B is distilled from DeepSeek-R1-14B at each temperature, with the reference bank fixed at T=1.0. Cells are the percentage of up to 100 distilled models flagged, at n-gram=3.

Temp.	WM-Basin			Black-box		
	TPR@FPR			TPR@FPR		
	5%	1%	.1%	5%	1%	.1%
1.0	100	99	88	23	9	1
0.7	99	93	62	12	2	0
0.5	71	51	22	4	2	0

Table 3: **Distillation detection across teacher models.** Qwen2.5-3B is distilled from three additional teachers at temperature $T = 1.0$, with the detector's reference bank rebuilt to match. Cells are read as in Tab. 2, at n-gram=3; `Acc.` is mean student accuracy.

Teacher	Acc.	WM-Basin			Black-box		
		TPR@FPR			TPR@FPR		
		5%	1%	.1%	5%	1%	.1%
DeepSeek-R1-14B	68.9	100	99	88	23	9	1
gpt-oss-120B	77.2	96	76	30	18	10	2
QwQ-32B	80.8	99	97	91	88	74	42
Llama-3.1-8B	74.9	100	100	100	93	87	64

Table 4: **Distillation detection across watermarking schemes.** The teacher's default TextSeal decoder is replaced by each of three other schemes, with the detector's reference bank rebuilt to match. Cells are read as in Tab. 2, over 50 distilled models per scheme and window.

Watermark	n-gram=1						n-gram=2						n-gram=3					
	WM-Basin			Black-box			WM-Basin			Black-box			WM-Basin			Black-box		
	TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR		
	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%
GumbelMax	100	100	100	72	44	28	100	98	98	52	32	6	100	100	100	22	10	0
KGW	100	100	100	58	26	8	94	86	56	40	20	4	90	68	34	16	2	2
SynthID	100	100	100	64	38	8	100	100	98	44	16	0	100	96	78	24	8	4

output distribution on its most likely tokens. For WM-Basin, the reference gradient bank remains fixed at T=1.0, creating a mismatch at the two lower temperatures. Table 2 shows that lowering the distillation temperature costs the black-box baseline most of its power at every window, whereas WM-Basin at $p \leq 0.01$ still flags every suspect at n-gram=1 and about half of them at n-gram=3 even at T=0.5. Hence, temperature governs *what the tokens reveal, not what the gradient reveals*: lowering it is a cheap evasion against the baseline and none against WM-Basin.

Effect of the teacher model. Are some teachers harder to protect than others? We distill the same Qwen2.5-3B student from each of four teachers spanning three families and 8B to 120B parameters: DeepSeek-R1-14B, gpt-oss-120B (Agarwal et al., 2025), QwQ-32B (Team, 2025), and Llama-3.1-8B (Grattafiori et al., 2024), rebuilding the reference bank for each. Tab. 3 shows that, at $p \leq 0.05$, WM-Basin retains its power on all four teacher models, detecting almost all distilled models. The baseline loses most of it on two teacher models and approaches ours on the remaining two. Thus, the baseline's power is inherited from the teacher and varies sharply with it, whereas WM-Basin is a more robust detector across all teacher models.

Effect of the watermark scheme. Is detection tied to the watermark the teacher happens to deploy? We replace TextSeal in the teacher's decoder with GumbelMax (Aaronson, 2022), KGW ($\gamma = 0.5$) (Kirchenbauer et al., 2023), and SynthID (Dathathri et al., 2024), rebuilding the reference bank for each and distilling 50 students per scheme and window. Table 4 shows that all three transfer: WM-Basin flags every student at $p \leq 0.01$ and n-gram=1 under each of them. It outperforms the baseline in every cell of the sweep, by a margin that widens with the window at $p \leq 0.05$. We note that WM-Basin needs only traces the teacher generated under a key, not how the watermark's own test works. The baseline needs the test itself, since it scores the suspect's tokens with it, and that test is derived scheme by scheme (§3.1). Nothing in WM-Basin is therefore specific to a particular scheme: to audit a new keyed watermark the verifier generates teacher traces under its keys, rebuilds the reference bank from them, and runs the same detector.

Effect of the audited domain and system prompt. How close must the audited domain be to the adversary's distillation data? We hold the 100 suspects fixed and change the domain the verifier audits, from math domain they were distilled on (OpenR1-Math, MATH, GSM8K) to two other domains outside it (code and chat), rebuilding the probe warm-up, the task subspace, and both detectors on

Table 5: **Distillation detection across the audited domain.** The same 100 suspects, all distilled on OpenR1-Math (Hugging Face, 2025), are audited on three math datasets from the distillation domain (top block) and two domains outside it (bottom block), with both detectors built entirely from the audited domain's own prompts and system prompt. Cells are read as in Tab. 1. Detailed setup in App. D.4.

Audited domain	n-gram=1						n-gram=2						n-gram=3					
	WM-Basin			Black-box			WM-Basin			Black-box			WM-Basin			Black-box		
	TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR		
	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%
OpenR1-Math	100	100	96	76	49	19	98	92	69	51	24	9	95	83	45	26	12	2
MATH	91	74	56	74	49	20	71	48	15	60	28	7	65	38	9	24	7	2
GSM8K	29	14	3	35	13	2	18	7	1	16	9	1	22	10	1	7	3	0
Code	12	5	0	50	19	4	5	1	0	31	11	2	8	4	0	7	2	0
Chat	11	1	0	22	4	1	2	0	0	15	4	0	4	1	0	13	4	1

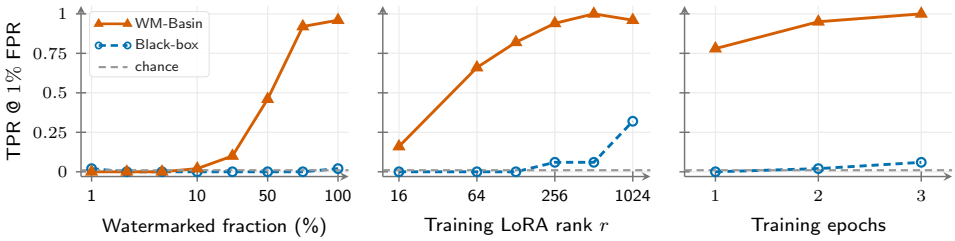


Figure 3: **What governs detection: each knob swept on its own axis** (n-gram=3). Each panel varies *one* knob, with its value on the x-axis and the true-positive rate at a fixed 1% false-positive rate on the y-axis (TPR@1% FPR; higher = more detectable), for our WM-Basin detector and the black-box baseline; the grey dashed line marks chance (TPR=FPR=1%). For each config we train 50 students, each with a *different* watermark key. Panel 1 shows the impact of the watermarked fraction, and Panels 2 and 3 the impact of LoRA rank r and training epochs. The all-clean control, with no watermarked traces, is measured but not plotted; none of its 50 students is flagged at any of the three thresholds.

each domain's own prompts, none of them the teacher's training prompt. All three windows are in Tab. 5, and the exact prompts and the per-domain rebuild are in App. D.4. WM-Basin's detection power ordering matches where the suspects were distilled: it ranks every math domain above both non-math ones in every cell, whereas the baseline ranks code above GSM8K at n-gram=1 and 2. A hit therefore tells the verifier which domain the suspect's text carries the watermark on, not only that it carries one; a low score elsewhere means nothing was found there, not that the model is clean (§3.3).

Effect of the watermarked fraction and the training recipe. What in the adversary's distillation data and distillation training recipe lets the watermark reach the student? At n-gram=3 we sweep one knob at a time around the default recipe: the watermarked fraction of a fixed 1,900-trace training set from 1 to 100%, with the remainder filled by clean traces from the same teacher, distillation LoRA rank r from 16 to 1,024, and training epochs 1, 2, and 3. Fig. 3 shows that detection power is mostly affected by the watermarked fraction and by the LoRA rank, and much less by how long the student trains: 1) WM-Basin performs at chance when at most 10% of the traces are watermarked and exceeds 90% TPR once 75% are; 2) its TPR climbs from 16% at r=16 to 100% at r=512; 3) even a single epoch yields 78% TPR. The black-box baseline (Sander et al., 2024) stays at or below 2% at every fraction and achieves its highest score in the figure at the largest rank, and WM-Basin outperforms it in all cells of the rank and epoch sweeps.

3.3 Post-processing and Basin Dynamics

In §2.2, we showed that a watermarked student sits in a narrow basin: a small step along the watermark direction (defined in Equation 1) leaves it, while a much larger step along the task direction does not. Next, we examine the direction in which standard post-processing moves the student and whether it removes the watermark.

Table 6: **Detection under compression.** Each row applies one operation to the same 100 distilled suspects and regenerates their text; *baseline* is uncompressed. Cells are read as in Tab. 2, at n-gram=3; `Acc.` is mean GSM8K, and rows below the mid-rule are under $0.9\times$ the baseline performance. `Along` w is the weight change projected on the watermark direction.

Compression	Acc.	Along w	WM-Basin TPR@FPR 5%	WM-Basin TPR@FPR 1%	WM-Basin TPR@FPR .1%	Black-box TPR@FPR 5%	Black-box TPR@FPR 1%	Black-box TPR@FPR .1%
int8	70.0	−0.0001	100	99	85	24	7	1
prune10	77.3	−0.0058	99	94	77	21	8	1
prune20	57.6	−0.0348	91	82	49	20	4	0
int4	45.4	−0.0117	46	25	10	7	1	0
nf4	41.2	−0.0266	95	79	40	8	3	0
prune30	28.9	−0.0887	49	26	7	14	1	0

Table 7: **Detection under continued training.** Each row continues training the same suspects on non-watermarked data; Δ`Acc.` is the change in GSM8K accuracy, and `Along` w is as in Tab. 6. Detector cells are read as in Tab. 2 (n-gram=3). **Open-weight** (Sander et al., 2024) is teacher-forced on the adversary's own traces with knowledge of exact training traces.

Cond.	ΔAcc.	Along w	WM-Basin TPR@FPR 5%	WM-Basin TPR@FPR 1%	WM-Basin TPR@FPR .1%	Black-box TPR@FPR 5%	Black-box TPR@FPR 1%	Black-box TPR@FPR .1%	Open-weight TPR@FPR 5%	Open-weight TPR@FPR 1%	Open-weight TPR@FPR .1%
Continued SFT on clean cross-domain data											
reason	+2.4	−0.146	6	1	0	7	3	1	98	95	73
multi.	+6.0	−0.007	21	4	0	15	5	0	100	100	97
Continued RL (GRPO)											
GSM8K	+0.5	0.000	100	93	53	0	0	0	100	100	100
K&K	−0.2	0.000	100	100	50	0	0	0	100	100	100
IFEval	−0.3	0.000	100	100	50	10	0	0	100	100	100

Effect of model compression. Can an adversary quantize or prune the watermark out of a deployed student? We apply int8, int4, nf4, or 10 to 30% unstructured pruning to the same 100 suspects. Tab. 6 shows that only int8 and 10% pruning leave a usable model, and both cost WM-Basin at most three points at any operating point. Other operations weaken detection but also leave the student unusable. Every operation displaces the student at least three orders of magnitude further across the watermark direction than along it. This confirms that compression does not move the student model along the watermark direction, so it remains in the watermark basin and is therefore detectable.

Effect of continued SFT and RL. Does continuing to train the student on clean data move the distilled model out of its watermark basin? We continue training the suspects on cross-domain data via SFT and GRPO (Shao et al., 2024) (training details are given in App. E.2), holding the probe and the reference bank fixed. Table 7 shows that continued RL leaves the student detectable, while continued SFT weakens the signal: WM-Basin detects all distilled models in every GRPO arm at $p \leq 0.05$, whereas both SFT arms substantially reduce its performance. This is consistent with the magnitude of the model's movement along the watermark direction: continued SFT on general reasoning traces moves the distilled model farther along this direction than SFT on multilingual data, whereas RL barely moves it. The open-weight detector nevertheless remains effective. This suggests that continued SFT weakens the signal observable in the distilled model's outputs without moving the model outside the watermark basin.

4 Conclusion and Future Work

In this work, we introduce the *watermark basin*: distilling on watermarked traces encodes the teacher's secret key as a faint direction in the parameter space, displacing the student weights along that key direction and into a region that later training does not easily escape. Building on this geometry, we design WM-Basin, which reformulates black-box detection of unauthorized distillation from a fragile, surface-level token count into a geometric alignment test in parameter space. Across a wide range of evaluations, spanning six student families, four teachers and four watermarking schemes, WM-Basin outperforms the existing baseline by a large margin, and it survives quantization, pruning and RL post-training.

Scope and outlook. A watermark can be present in a distilled model but unreadable in its text: continued SFT on clean data suppresses the text-level signal while an open-weight probe still recovers the residual, and a supplementary study reported in App. E.3 finds the same for rewriting the teacher's traces before distillation. WM-Basin is also an in-domain test: a hit tells the verifier which domain the suspect's text carries the watermark on, not only that it carries one; a low score elsewhere means nothing was found there, not that the model is clean (§3.2). Both are limits on what a verifier can observe, not on what distillation transfers. Further narrowing the gap between the black-box read and the open-weight probe is thus a promising future direction.

AI USE STATEMENT

We used generative AI tools for two purposes in this work: writing and debugging research code, including the distillation and evaluation pipelines, and the plotting scripts, and editing the manuscript to improve readability. We did not use generative AI tools to develop the conceptual or theoretical framework, to formulate or prove mathematical claims, to propose or refine hypotheses, to design the methodology or the experiments, or to interpret results. We have reviewed all AI-assisted work and take responsibility for the final content of this paper, including its text, claims, and results.

ETHICS STATEMENT

All authors have read the ICLR Code of Ethics and adhere to it. This work studies whether a model provider can establish, from API access alone, that a suspect model was distilled from its watermarked teacher. It involves no human subjects and no personal or user data: the distillation corpus (OpenR1-Math-220k), the utility benchmarks (GSM8K and Math-500), and every teacher, student, probe, and clean-pool checkpoint are publicly released artefacts used under their own licenses, and every experiment runs on models we host ourselves.

Because the method produces evidence of provenance, we report how it behaves on models that are not guilty rather than a bare score. The paper also measures what weakens the audit, including rewriting the teacher's traces before distillation, continued fine-tuning, and probing away from the distillation domain. These are variants of attacks already described in the text-watermarking literature, and we report them because a verifier that does not know the limits of its own test is the more likely harm.

REPRODUCIBILITY STATEMENT

The code is included in the supplementary material. The configuration behind every number in the paper is recorded in App. C. Table 8 gives the default models, watermark and sampling settings, distillation data and LoRA hyperparameters, probe warm-up, and detector and baseline settings in full, down to the key ranges and seeds; Table 9 indexes every study, the factor it varies, the cohort and sample size used, and where its result is reported, so a single arm is reproduced by reading the defaults plus the one row that changes them. The detector is specified procedurally in Alg. 1. On the data side, App. C.2 documents trace generation and the quality filter that takes $5{,}000$ sampled traces to the $1{,}900$ used for training.

REFERENCES

Scott Aaronson. My AI safety lecture for UT effective altruism, nov 2022. URL https://scottaaronson.blog/?p=6823. Blog post. Accessed: 2026-03-17.

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX security symposium (USENIX Security 18)*, pp. 1615–1631, 2018.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

Anthropic. Detecting and preventing distillation attacks, feb 2026a. URL https://www.anthropic.com/news/detecting-and-preventing-distillation-attacks. Accessed: 2026-02-25.

Anthropic. How Claude marks AI-generated content, aug 2026b. URL https://support.claude.com/en/articles/16266773-how-claude-marks-ai-generated-content. Accessed: 2026-08-13.

Huanran Chen, Zeming Wei, Yao Huang, Yichi Zhang, Yinpeng Dong, and Jun Zhu. Unveiling the basin-like loss landscape in large language models. In *International Conference on Learning Representations*, volume 2026, pp. 148112–148146, 2026.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7001–7016, 2024.

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

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*, 2014.

Meyer Dwass. Modified randomization tests for nonparametric hypotheses. *The Annals of Mathematical Statistics*, pp. 181–187, 1957.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 55–65, 2019.

European Commission. Code of Practice on Transparency of AI-Generated Content, 2026. URL https://digital-strategy.ec.europa.eu/en/faqs/code-practice-transparency-ai-generated-content.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. In *2023 IEEE international workshop on information forensics and security (WIFS)*, pp. 1–6. IEEE, 2023.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*, 2019.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

Google Threat Intelligence Group. GTIG AI threat tracker: Distillation, experimentation, and (continued) integration of AI for adversarial use, feb 2026. URL https://cloud.google.com/blog/topics/threat-intelligence/distillation-experimentation-integration-ai-adversarial-use. Accessed: 2026-02-26.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

William B Johnson, Joram Lindenstrauss, et al. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International conference on machine learning*, pp. 17061–17084. PMLR, 2023.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Xinhang Ma, William Yeoh, Ning Zhang, and Yevgeniy Vorobeychik. Protecting language models against unauthorized distillation through trace rewriting. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11307–11324, 2026.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. *arXiv preprint arXiv:2104.10706*, 2021.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*, 2017.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

Leyi Pan, Aiwei Liu, Shiyu Huang, Yijian Lu, Xuming Hu, Lijie Wen, Irwin King, and Philip S Yu. Can llm watermarks robustly prevent unauthorized knowledge distillation? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13228–13251, 2025.

Alexander Panfilov, David Schmotz, Ilia Shumailov, Luca Beurer-Kellner, Joachim Schaeffer, Ameya Prabhu, Jonas Geiping, and Maksym Andriushchenko. Stealing reasoning traces from proprietary llm apis. *arXiv preprint arXiv:2608.09867*, 2026.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023.

Sheng Y Peng, Pin-Yu Chen, Matthew Hull, and Duen H Chau. Navigating the safety landscape: Measuring risks in finetuning large language models. *Advances in Neural Information Processing Systems*, 37:95692–95715, 2024.

Fortunato Pesarin and Luigi Salmaso. *Permutation tests for complex data: theory, applications and software*. John Wiley & Sons, 2010.

Belinda Phipson and Gordon K Smyth. Permutation p-values should never be zero: calculating exact p-values when permutations are randomly drawn. *arXiv preprint arXiv:1603.05766*, 2016.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of machine learning research*, 11(sept):2487–2531, 2010.

Daniel Remler and Ben Hayum. Adversarial distillation. Center for a New American Security (CNAS) Report, jun 2026. URL https://www.cnas.org/publications/reports/adversarial-distillation. Accessed: 2026-08-01.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Radioactive data: tracing through training. In *International Conference on Machine Learning*, pp. 8326–8335. PMLR, 2020.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. *Advances in Neural Information Processing Systems*, 37: 21079–21113, 2024.

Tom Sander, Hongyan Chang, Tomáš Souček, Tuan Tran, Valeriu Lacatusu, Sylvestre-Alvise Rebuffi, Alexandre Mourachko, Surya Parimi, Christophe Ropers, Rashel Moritz, et al. Textseal: A localized llm watermark for provenance & distillation protection. *arXiv preprint arXiv:2605.12456*, 2026.

Yash Savani, Asher Trockman, Zhili Feng, Yixuan Xu, Avi Schwarzschild, Alexander Robey, Marc Finzi, and Zico Kolter. Antidistillation sampling. *Advances in Neural Information Processing Systems*, 38:117800–117827, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https://qwenlm.github.io/blog/qwq-32b/.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pp. 601–618, 2016.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, pp. 269–277, 2017.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2742–2785, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Tingwei Zhang, John X Morris, and Vitaly Shmatikov. How to steal reasoning without reasoning traces. *arXiv preprint arXiv:2603.07267*, 2026.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. Distillation-resistant watermarking for model protection in nlp. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5044–5055, 2022.

Appendix Outline

A Extended Related Work **14**

B Watermark Basins and Algorithm **15**

B.1 The Key-Randomization Rank Test 15

B.2 The WM-Basin Pipeline in Pseudocode 16

C Experimental Setup **17**

C.1 Configuration Summary 17

C.2 Distillation and Student Training 19

C.3 Prompt Pools and Suspect Sampling 19

C.4 Detector and Baseline Implementation 20

C.5 Removing Shared Task Directions 21

C.6 Calibration of the Rank p-Value 22

D Additional Detection Results **23**

D.1 Operating Points and Distillation Temperature 23

D.2 Cross-Teacher Detection 23

D.3 Watermark Scheme 25

D.4 Audited Domain and System Prompt 26

D.5 Suspect-Text Budget 27

D.6 Probe Model and Warm-Up 27

E Additional Robustness Results **28**

E.1 Compression 28

E.2 Continued Fine-Tuning and RL 29

E.3 Paraphrasing 32

A Extended Related Work

Model extraction and unauthorized distillation. Frontier providers, threat-intelligence groups, policy analysts, and security researchers now document distillation-oriented misuse in the wild: adversaries harvest reasoning traces at scale from deployed APIs to train competing models (Anthropic, 2026a; Google Threat Intelligence Group, 2026; Panfilov et al., 2026). The underlying mechanism is knowledge distillation, which transfers capability from a teacher to a student model (Hinton et al., 2015). When exposed through an API, it is called *model extraction*, where black-box query access alone, without the target's parameters or training data, suffices to replicate the teacher model functionality (Tramèr et al., 2016). The threat is sharpest for reasoning models, whose step-by-step traces provide dense supervision. We study supervised fine-tuning on collected reasoning traces, motivated by evidence that small, high-quality reasoning sets sharply improve students (Muennighoff et al., 2025).

Watermarking, radioactivity, and ownership resolution. Defenses against extraction either restrict what the API returns, monitor extraction-like querying (Anthropic, 2026a; Google Threat Intelligence Group, 2026), or make outputs less useful for training via antidistillation sampling or trace rewriting (Savani et al., 2026; Ma et al., 2026). Such methods trade away utility for benign users

or assumes the attacker stays observable during collection. In practice, this assumption is fragile: adversaries spread collection across many accounts and third-party API resellers, so no single query stream looks like extraction Anthropic (2026a). Watermarking is a complementary defense: it embeds a secret statistical signal into generated text (Kirchenbauer et al., 2023; Kuditipudi et al., 2023; Sander et al., 2026) and later tests whether a suspect model inherited it (Sander et al., 2024). The idea of tracing data through a trained model originates in the computer vision literature. Sablayrolles et al. (2020) show that imperceptible "radioactive" watermarks in a dataset leave a statistical residue in models trained on it, detectable with a calibrated p-value even when only 1% of the data is marked. Our rank test follows the same framework—Sander et al. (2024) adopts the idea of radioactivity into language models, showing watermarked distillation leaves a key-specific residue in the student, and others study how well such watermarks survive distillation (Sander et al., 2026; Zhao et al., 2022; Pan et al., 2025). A parallel black-box line, dataset inference (Maini et al., 2021), resolves ownership by statistically testing whether a suspect shares private knowledge of the victim's data without inserting task-disruptive watermarks, and observes that such insertions are off-distribution and therefore not always preserved under stealing.

The black-box verifier of Sander et al. (2024), which counts watermark-favored tokens in a suspect's sampled text, defines the setting we work in and is, to the best of our knowledge, the only existing method for the task. It is the baseline we compare against throughout. Its limits were not previously characterised, and establishing them is also part of this paper's contribution. The statistic functions only where exact watermarked contexts recur, which we show is fragile to wider windows, lower decoding temperatures, and larger text budgets (§3.2). This is the mode of failure that paraphrase attacks exploit against surface-level watermark detection more broadly (Sadasivan et al., 2023). WM-Basin keeps the access level and the generations and changes the statistic to a weight-space one. Classical model watermarking instead hides watermarks directly in weights or via backdoors (Uchida et al., 2017; Adi et al., 2018), read either from the parameters or by querying a trigger set; both assume the owner watermarked the suspect model under test, whereas the suspect we consider in this work is a model the owner never trained.

Weight-space geometry and basins. WM-Basin benefits from the fact that all students distilled under one key settle into a shared, key-specific region of weight space. This connects to the geometry of neural loss landscapes: independently trained optima are joined by simple, near-constant-loss curves (Garipov et al., 2018). Once permutation symmetries are accounted for the landscape resembles a single wide *basin*, so separately trained models can be aligned by permutation and merged in weight space (Ainsworth et al., 2022). Large language models exhibit the same basin-like structure (Chen et al., 2026; Li et al., 2018; Peng et al., 2024). WM-Basin treats the watermark key as carving out one such basin and asks whether a suspect has fallen into it.

Gradient attribution and representation post-processing. To read this geometric signal from text alone, WM-Basin backpropagates a suspect's completions through a fixed probe model. Intuitively, a training example's gradient captures how it would have nudged the model, so aligned gradients means they are obtained from similar examples. However, raw gradient cosines are dominated by a few shared directions (generic language and formatting)—a handful of top components inflate similarity between otherwise unrelated items (Ethayarajh, 2019; Gao et al., 2019; Radovanovic et al., 2010; Dinu et al., 2014). We remove these with an "All-but-the-Top" projection (Mu et al., 2017), stripping the few dominant yet uninformative directions so that the faint key-specific watermark basin stands out from generic task geometry.

B Watermark Basins and Algorithm

We explain §2 in more details: Fig. 4 is a more detailed presentation of the watermark basin of Fig. 1; App. B.1 states the rank test that turns a score into a p-value, and App. B.2 explains the detector algorithm step by step.

B.1 The Key-Randomization Rank Test

Token-level detectors read significance from a parametric null for their score: a Gaussian or binomial tail for a green-list count (Kirchenbauer et al., 2023; Sander et al., 2024), a Gamma tail for a Gumbel one (Fernandez et al., 2023), which is the form our black-box baseline uses (App. C.4). WM-Basin

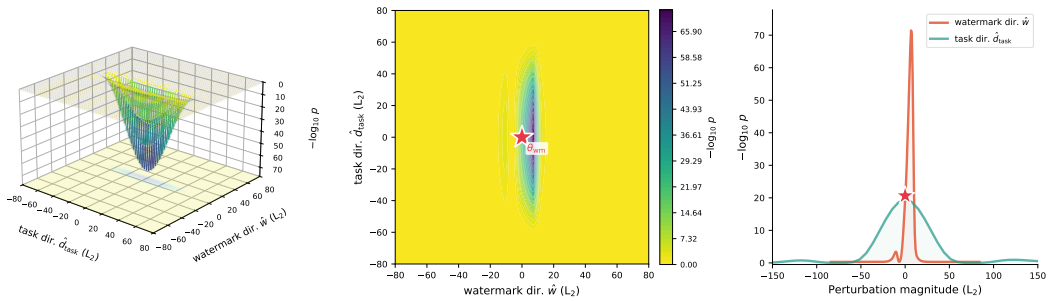


Figure 4: **The anisotropic geometry of the *watermark basin*** measured as the white-box detection significance ($-\log_{10} p$) around a radioactive student $\boldsymbol{\theta}_{\text{wm}}$ (red star) fine-tuned on watermarked teacher traces. **Left to right:** 3D landscape; 2D contour map; and 1D slices through $\boldsymbol{\theta}_{\text{wm}}$. We observe that the basin is asymmetric: the signal falls to the detector's floor within a few steps along the watermark direction $\hat{\Delta}_{\text{wm}}$ (red curve), while remaining detectable over a much broader range along the task direction $\hat{\Delta}_{\text{task}} = \Delta_{\text{task}}/\|\Delta_{\text{task}}\|_2$ (green curve).

assumes no such model for S and reads significance from the rank of the claimed key among M decoy keys drawn from the same generator (Dwass, 1957; Phipson & Smyth, 2016).

Let H_0 be the null hypothesis of §2.1: the suspect's outputs on the audited domain carry no information about the claimed key $k^\star$, as holds when the suspect was never trained within that key's basin. To test it, we fix the suspect model, the prompts and the detection pipeline, then draw M independent decoy keys $k_1, \ldots, k_M \leftarrow \mathsf{KeyGen}$. We evaluate S at $k^\star$ and at each decoy, where a higher score means stronger basin alignment, and form the empirical p-value of Eq. (3): how uniquely the suspect aligns with $k^\star$ relative to random basins. The leading 1 counts $k^\star$ among the $M{+}1$ candidates, which prevents $p_{\text{rank}}{=}0$ and keeps Type-I error control valid under finite Monte Carlo sampling (Phipson & Smyth, 2016).

The p-value is valid because nothing in the pipeline can favor the claimed key under H_0. The claimed key and the M decoys are drawn the same way, and the scoring rule uses a key only as an index that selects reference traces; the probe, the random projection, the build prompts and the fitted U are identical for all $M{+}1$ candidates. The claimed key is therefore equally likely to occupy any of the $M{+}1$ ranks, assuming no ties. Thresholding its rank p-value then controls the FPR in finite samples: the FPR is at most α for any threshold, and equals α when α lies on the attainable rank grid. The $\geq$ in Eq. (3) counts ties against rejection, so they cost power rather than validity. None of this needs a parametric model of S (Pesarin & Salmaso, 2010): Eq. (3) places no restriction on the score, so the guarantee survives arbitrary non-linear transformations of the suspect model. App. C.6 measures the realized rate against the nominal one for every deployed key.

The smallest attainable p-value is $1/(M{+}1)$, so the decoy count sets the finest significance an audit can certify: reporting at $\alpha{=}0.001$ requires $M \geq 999$ decoys. Decoy scoring is offline and reuses the build-once artefacts, so improving the resolution can be cheap in practice.

B.2 The WM-Basin Pipeline in Pseudocode

The whole detection procedure is presented in Algorithm 1. The **Build once** block never touches the suspect. It reads only the candidate keys, the build pool $\mathcal{X}_{\text{build}}$ and the pool of clean checkpoints. We warm the probe θ_{probe} with a short LoRA pass on unwatermarked teacher traces, then backpropagate each candidate key's teacher traces through the probe to obtain that key's reference gradient Δ_k (stage (i)). We fit the shared task basis U by PCA on the clean pool's probe gradients and project it out of every reference gradient (stage (ii)).

The **Detect** block is the only part paid per suspect. We sample the suspect on the test pool $\mathcal{X}_{\text{test}}$, take the gradient of that text at the same probe, and project it off the same U. Scoring each candidate key with the residual cosine of Eq. (2) and ranking the claimed key against the decoys returns the p-value of Eq. (3) (stage (iii)). A verifier therefore builds the reference bank and U once and reuses them across audits.

Algorithm 1 WM-Basin: Black-Box Detection of Watermark Distillation via Basin Alignment

Require: Query access to suspect model $\mathcal{M}_{\text{sus}}$ (black-box API) and watermarked teacher; Claimed production key $k^\star$ and M decoy keys $k_1, \ldots, k_M \leftarrow \mathsf{KeyGen}$; Disjoint build and test prompt pools $\mathcal{X}_{\text{build}}$, $\mathcal{X}_{\text{test}}$ from the audited domain D; Pool of clean off-the-shelf models; Significance level α.
Ensure: p-value p_{rank}; report suspect as distilled into the basin of $k^\star$ when $p_{\text{rank}} \leq \alpha$.

Build once *(Offline; independent of suspect)*
$\theta_{\text{probe}} \leftarrow$ clean small model warmed by a short LoRA pass on unwatermarked teacher traces from $\mathcal{X}_{\text{build}}$.
for each key $k \in \{k^\star, k_1, \ldots, k_M\}$ **do**
  Sample teacher responses under key k on $\mathcal{X}_{\text{build}}$.
  $\Delta_k \leftarrow$ average gradient at probe θ_{probe} ▷ Raw reference gradient for key k
end for
Form task basis $U = [u_1, \ldots, u_q]$ holding top-q principal directions of clean pool gradients at θ_{probe} on $\mathcal{X}_{\text{build}}$.
for each key $k \in \{k^\star, k_1, \ldots, k_M\}$ **do**
  $\widetilde{\Delta}_k \leftarrow (I - UU^\top)\Delta_k$ ▷ Residual reference watermark gradient
end for

Detect *(Online; once per suspect)*
Sample answers from suspect model $\mathcal{M}_{\text{sus}}$ on $\mathcal{X}_{\text{test}}$. ▷ Unused by every step above
$\Delta_{\text{sus}} \leftarrow$ average gradient of suspect text backpropagated through probe θ_{probe}.
$\widetilde{\Delta}_{\text{sus}} \leftarrow (I - UU^\top)\Delta_{\text{sus}}$ ▷ Residual suspect gradient
for each key $k \in \{k^\star, k_1, \ldots, k_M\}$ **do**
  $S(k) \leftarrow \cos\left(\widetilde{\Delta}_{\text{sus}}, \widetilde{\Delta}_k\right)$ ▷ Residual basin alignment score
end for
$p_{\text{rank}} \leftarrow \frac{1+\sum_{j=1}^{M} \mathbb{I}\left(S(k_j) \geq S(k^\star)\right)}{M+1}$ ▷ Non-parametric key-randomization p-value
return p_{rank}; report "distilled into basin of $k^\star$" if $p_{\text{rank}} \leq \alpha$.

C Experimental Setup

This section presents the experiment setup in §3.

C.1 Configuration Summary

Table 8 gives in full the default models and numeric settings that §3.1 states in summary, and Tab. 9 indexes the studies built on them, what each varies and where it is reported.

Table 8: **Default experimental configuration.** Models, watermark, distillation, probe, and detector settings used throughout §3 unless stated otherwise; see App. C.2 for full details. The TextSeal parameter α=0.1 is held fixed in every experiment.

Component	Setting
Models	
Teacher (default)	DeepSeek-R1-Distill-Qwen-14B
Teacher (swaps)	gpt-oss-120B, QwQ-32B, Llama-3.1-8B
Student (default)	Qwen2.5-3B
Student (cross-arch.)	24 structures, 135M to 8.2B (Llama(-Guard), Qwen2.5/3, QwQ, DeepSeek-Distill, Gemma, SmolLM2)
Probe model	Qwen2.5-3B (base, warmed)
Clean PCA pool	43 off-the-shelf checkpoints (15 families, 135M to 120B; 129 cells total; Tab. 10)
Watermark & teacher sampling	
Scheme	TextSeal, α=0.1
Sampling	vLLM; T=1.0, top-p=0.95; ≤8,192 tokens
Distillation temperatures	$\{1.0, 0.7, 0.5\}$ (reference fixed at T=1.0)
Distillation (student)	
Dataset	OpenR1-Math-220k
System prompt	one fixed math-teacher prompt for every trace; by default every verifier stage (warm-up, bank, clean pool, audit) renders this same prompt, so the default audit is prompt-matched. Only the domain sweep replaces it (App. D.4)
Traces	5,000 sampled → quality-filtered → 1,900 used for training

Table 8, continued.

Component	Setting
Trace filter	closing `</think>`, `\boxed{}`, no $\geq$3$\times$ repetition, `math_verify` match
Fine-tuning	LoRA r=512, lr 2e$-$5, 3 epochs, effective batch 16, max sequence length 8192
LoRA targets	$\{q, k, v, o, \text{gate}, \text{up}, \text{down}\}_{\text{proj}}$
Eval benchmarks	GSM8K, Math-500
Probe warm-up	
Data	450 clean teacher traces, on build-pool prompts $\mathcal{X}_{\text{build}}$ from the domain being audited (App. D.4)
Adapter	LoRA r=128, 4 epochs, lr 2e$-$4, batch 1
Detector (WM-Basin)	
Reference bank	100 deployed keys, one student distilled per key; each ranked against a 1,000-key bank ($k^\star$ + 999 decoys)
Per-key gradient	average of 512 traces ($\geq$512 tokens) on build-pool prompts $\mathcal{X}_{\text{build}}$
Projection	fixed random projection to 8,192-dim (TRAK)
Task subspace	top-q=100 PCA (all-but-the-top), fit on clean pool
p-value floor	0.001
Operating point	threshold $p\leq$0.01 (TPR@1% FPR)
Baseline (Sander et al., 2024)	
Black-box n-gram	corpus φ from keyed teacher traces on 5,000 held-out prompts
Windows	n-gram $\in \{1, 2, 3\}$
Deduplication	each distinct (context, token) pair scored once across traces
Test	baseline's own analytic p-value: two-key score fused at the generation α=0.1, variance-corrected Gamma tail (App. C.4); plain, not entropy-weighted
Suspect text (shared by every detector)	
Budget	1,900 held-out prompts $\rightarrow$ median 1,875 scored responses, $\approx$10M sampled tokens of which $\approx$1M are scored (App. C.3)
Prompts	test pool $\mathcal{X}_{\text{test}}$, disjoint from $\mathcal{X}_{\text{train}}$ and $\mathcal{X}_{\text{build}}$ (App. C.3)
System prompt	the distillation prompt above, i.e. matched; replaced only in App. D.4
Sampling	T=1.0, top-p=0.95, $\leq$8,192 new tokens, repetition penalty 1.0, seed 42

Table 9: **Per-experiment settings.** What each study varies, the cohort used, and where the result is reported; varied factors are read against the defaults of Tab. 8.

Experiment	Varied factor and levels	Model cohort / sample size	Result
Student architecture	24 structures, 135M to 8.2B parameters	50 students per architecture (1,200 total)	Tab. 1
Distillation temperature	Teacher sampling $T \in \{1.0, 0.7, 0.5\}$; reference bank fixed at T=1.0	100 students per temperature	Tabs. 2, 12
Teacher model	4 teachers spanning 3 families, 8B to 120B; bank rebuilt per teacher; the three swapped teachers distil for 8 epochs, not 3	100 students per teacher	Tabs. 3, 13
Watermark scheme	4 watermark schemes at n-gram $\in \{1, 2, 3\}$	50 students per scheme and window	Tabs. 4, 14
Audited domain	5 domains; every detector stage and the system prompt rebuilt per domain	Fixed pool of 100 students	Tab. 5
Model compression	int8, int4 and nf4 quantization; 10 to 30% pruning	100 students per operation	Tabs. 6, 17, 18; Fig. 10
Continued SFT and RL	Post-distillation training: SFT domain, or GRPO task	100 students (SFT); 10 (GRPO)	Tabs. 7, 19, 20, 21
Probe model	10 probe checkpoints; reference bank rebuilt per probe	Fixed pool of 100 students	Tab. 15
Probe warm-up	Presence, corpus domain, LoRA rank, learning rate, epochs; one knob at a time, n-gram=3 only	Fixed pool of 100 scored keys, reused across all 15 arms	Tab. 16
Trace paraphrasing	Same-family Qwen paraphraser at 3 strengths	20 students per strength	Tab. 22
Watermarked data fraction	Proportion f of watermarked traces, at a fixed 1,900-trace training budget	50 students per fraction	Fig. 3
Student training recipe	LoRA rank r, epochs	50 students per configuration	Fig. 3

Table 9, continued.

Experiment	Varied factor and levels	Model cohort / sample size	Result
Distillation utility	Clean vs. watermarked teacher traces	100 watermarked students per n-gram window, plus a clean control	Fig. 5
Operating-point ROC	Detection threshold, at n-gram $\in \{1, 2, 3\}$	100 watermarked students against 6,500 null cases	Fig. 8
Suspect-text budget	Cumulative prefixes of 5 to 1,878 generations (0.03 to 10.3M tokens supplied)	Fixed pool of 100 students	Fig. 9
Per-key false-positive rate	Which bank key is read as the deployed one, at α=5% and n-gram $\in \{1, 2, 3\}$	All 1,000 bank keys, each across 129 held-out clean model-temperature cells	Tab. 11
Watermark basin geometry	Weight-space offset along $\hat{\Delta}_{\text{wm}}$ and $\hat{\Delta}_{\text{task}}$	One student distilled under a single key	Fig. 4
Task-subspace removal	Removed components q: seven settings, from *no PCA* through *mean only* to q=100	1,200 distilled students at all n-grams, against 129 clean model-temperature cells	Fig. 6
Rank p-value calibration	Significance threshold α, swept under the null	100 deployed keys across 65 held-out clean model-temperature cells	Fig. 7

C.2 Distillation and Student Training

The student follows the fine-tuning recipe of Tab. 8. A teacher trace is retained only if it contains a closing `</think>` tag and a `\boxed{...}` answer, has no substring repeated $\geq$3 times under the sliding 100-character repetition check, and its extracted answer matches the gold answer under `math_verify`. The probe θ_{probe} is warmed up on clean teacher traces with LoRA.

Distillation performance. Watermarking the teacher does not cost the student the utility distillation buys. Figure 5 compares clean and TextSeal distillation under this recipe on GSM8K (Cobbe et al., 2021) and Math-500 (Hendrycks et al., 2021): distillation lifts both benchmarks well above the un-distilled base, and the watermarked students match the clean-distilled control at trace-filter pass rates within 8 points of it.

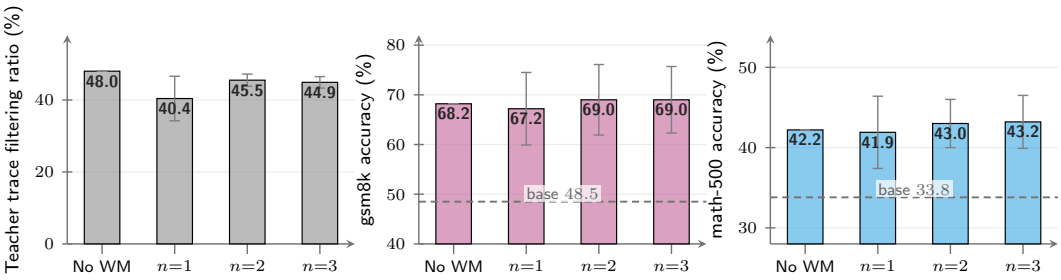


Figure 5: **Watermarking preserves the utility gained through distillation.** DeepSeek-R1-Distill-Qwen-14B (Guo et al., 2025) is distilled into Qwen2.5-3B (Yang et al., 2024) using clean traces or TextSeal traces at three n-gram windows. The left panel reports the fraction of 5,000 traces passing the quality filter; the other panels report GSM8K and Math-500 accuracy. Watermarked bars average 100 keyed students with $\pm$1 standard-deviation error bars, while No WM is one student trained on clean traces; dashed lines mark the un-distilled base.

C.3 Prompt Pools and Suspect Sampling

Section 2.1 splits the audited domain into disjoint training, build and test pools. In the following, we explain how we split the prompts in our evaluation.

- The *training* pool $\mathcal{X}_{\text{train}}$ is the adversary's: OpenR1-Math-220k prompts sampled under the fixed math-teacher system prompt, yielding $5{,}000$ traces, of which 40 to 48% pass the filter of App. C.2 (Fig. 5) and $1{,}900$ become the distillation set.
- The *build* pool $\mathcal{X}_{\text{build}}$ is the verifier's offline side and supplies the three artefacts WM-Basin is made of: the $\approx$450 clean teacher traces that warm the probe, the 512 keyed traces averaged into each of the $1{,}000$ reference directions, and the free generations of the 43 clean checkpoints that fit the task subspace. The baseline's n-gram corpus φ is a fourth offline artefact, drawn by construction from held-out teacher traces the suspect is never read on.
- The *test* pool $\mathcal{X}_{\text{test}}$ is $1{,}900$ held-out prompts and is the only pool a verdict is ever read on. No prompt the adversary trained on is used to build either detector, and no prompt used to build either detector is used to score a suspect.

C.4 Detector and Baseline Implementation

This section gives the implementation behind the detector and baseline of §3.1.

Reference key gradient bank and projection. For each candidate key, we generate teacher traces on the build pool $\mathcal{X}_{\text{build}}$ (App. C.3), discard responses below the minimum token length in Tab. 8, and average the surviving reference gradients into Δ_k. A single fixed random projection, shared by all keys and applied unchanged to the suspect gradient Δ_{sus} so that every cosine the rank test compares is taken in one coordinate space, approximately preserves their inner products (Johnson et al., 1984; Xia et al., 2024). We compute this with TRAK (Park et al., 2023). The deployed key and i.i.d. decoys form the bank used by the rank test in Eq. (3), and its size sets the reported p-value floor.

Compute cost. Nearly all of the cost is one-time and suspect-independent. Warming the probe is one short LoRA pass (App. D.6). Building the bank dominates: each projected direction averages 512 reference traces at about 27 minutes of single-GPU time, so around 450 GPU-hours for the full $1{,}000$-key bank, which occupies 31.2 MiB once projected. Both artefacts are reused by every subsequent audit and are rebuilt only when what they describe changes, namely the teacher or the probe for the bank, and the probe or the prompt distribution for the task subspace (App. C.5). Auditing one suspect then costs sampling its text, the $\approx$1M sampled tokens of App. C.3, a query cost identical for every detector, plus one backward pass and projection per completion; at the throughput above, the median $1{,}875$-completion budget is roughly 1.6 GPU-hours on one device.

Sampled-token baselines. The baselines of Sander et al. (2024) score a single token per position. In the open-weight setting, teacher-forced reading evaluates the student's top-1 prediction on teacher-generated test text. Their black-box variant scores tokens sampled by the student. We score a token only if its preceding n-gram appears in held-out teacher text generated with the key being tested. We exclude tokens whose preceding n-gram also appears in the prompt. Following Sander et al. (2024), both variants score distinct (context, token) pairs across traces. The open-weight variant additionally requires distinct contexts within each trace.

Seeing only text, the verifier re-tokenizes each suspect response with the *teacher's* tokenizer and scores every position of that stream: the context n-gram c_i and the scored token y_i both come from the suspect's own tokens, with no teacher forcing and no student logits. TextSeal's secret is a key pair (k_a, k_b), and the generator draws its Gumbel noise from k_a with probability α and from k_b otherwise, so no token reveals which key drove it. We score both and weight them by that generation prior,

$$S_i \;=\; \alpha\big(-\log(1-r_i^{(a)})\big) \;+\; (1-\alpha)\big(-\log(1-r_i^{(b)})\big), \qquad r_i^{(\cdot)} \;=\; \mathrm{PRF}\big(c_i,\, y_i;\, k.\big), \tag{4}$$

at the deployed α=0.1. Under H_0 each $-\log(1-r) \sim \mathrm{Exp}(1)$ and the two keys are independent, so $\mathbb{E}[S_i]$=1 but $\mathrm{Var}[S_i]$=$\sigma_\alpha^2 \equiv \alpha^2$+$(1-\alpha)^2$=0.82. Matching $S = \sum_{i=1}^{L} S_i$ over the L retained positions to a Gamma of mean L and variance $L\sigma_\alpha^2$ (shape L/σ_α^2, scale σ_α^2) gives

$$p \;=\; Q\!\left(\frac{L}{\sigma_\alpha^2},\, \frac{S}{\sigma_\alpha^2}\right), \qquad z \;=\; \frac{S-L}{\sigma_\alpha\sqrt{L}}, \tag{5}$$

with Q the upper regularized incomplete gamma function.

The three other watermark schemes in App. D.3 are read on the same principle: GumbelMax is the single-key case, where α=1 leaves σ_α^2=1, whereas GreenList and SynthID score each position with a keyed-vocabulary indicator and a tournament g-value rather than a Gumbel variate.

C.5 Removing Shared Task Directions

Section 2.3 projects the shared task subspace out of every gradient before scoring; this section gives how that subspace is fit and how many components we remove.

The idea of removing the shared task direction is aligned with removing the shared common vector for the word embeddings studied in Mu et al. (2017). Following their postprocessing, we fit PCA to gradients from the clean checkpoint pool and keep its top q principal directions as the shared task basis U, then project Δ_{sus} and every Δ_k onto the orthogonal complement of U before computing the cosine similarity score $S(k)$ in stage (iii). The number of removed directions, q, is the one free parameter of this procedure.

Table 10: **The clean PCA pool.** The 43 off-the-shelf checkpoints whose gradients fit the shared task subspace of App. C.5, grouped by pretraining lineage. Each model generates on 1,000 build-pool prompts at each of three decoding temperatures (0.7, 1.0, 1.3).

Family	Checkpoints	Ckpts.
Qwen	Qwen2.5-0.5B, -0.5B-Instruct, -1.5B, -1.5B-Instruct, -3B, -3B-Instruct, -7B-Instruct, -Math-7B; Qwen3-0.6B, -1.7B, -4B, -4B-Instruct, -8B, -8B-Base, -32B; QwQ-32B; Qwen1.5-MoE-A2.7B	17
Llama	Llama-2-7B, Llama-3.1-8B-Instruct, Llama-3.2-1B-Instruct, Llama-Guard-3-1B, Llama-Guard-3-8B	5
DeepSeek	DeepSeek-R1-Distill-Qwen-1.5B, -7B, -14B†; DeepSeek-MoE-16B-Base	4
SmolLM2	SmolLM2-135M, -135M-Instruct, -1.7B	3
Gemma	Gemma-2-2B, Gemma-3-1B-PT	2
Phi	Phi-3.5-mini-Instruct, Phi-4	2
OLMo	OLMo-2-1B, OLMoE-1B-7B	2
Mistral	Mistral-7B-Instruct	1
Yi	Yi-1.5-9B-Chat	1
Falcon	Falcon3-7B-Instruct	1
Granite	Granite-3.1-8B-Instruct	1
StarCoder	StarCoder2-7B	1
Nemotron	Nemotron-Mini-4B	1
Moonlight	Moonlight-16B-A3B	1
gpt-oss	gpt-oss-120B	1
15 families		43

What the clean pool is. Table 10 lists all models used for removing the task direction: 43 checkpoints are off-the-shelf, never-distilled causal LMs spanning 15 pretraining families and 135M to 120B parameters. None has been fine-tuned on the audited domain and none has ever been exposed to a watermark key. Every checkpoint contributes one cell per decoding temperature (0.7, 1.0, 1.3), giving the 129 cells of Tab. 8. A cell free-generates on the build pool $\mathcal{X}_{\text{build}}$, and its gradients are taken at the same probe model θ_{probe}, so a clean cell and a suspect differ only in the text they produced. For fixed-threshold TPR evaluation, clean models are not evaluation models, so the PCA fit uses all 129 cells. The ROC curves and the per-key calibration curves use the balanced subset, i.e., 43 checkpoints at three temperatures split into disjoint fit and evaluation halves.

How many components to remove. We choose q on clean models alone, and we read the false-positive rate of each single deployed key. Figure 6 sweeps q from *no PCA* to q=100 and reads three keys of the 1,000-key bank against TPR: the typical key, the key at the 95th percentile of the per-key rate in terms of FPR, and the worst key which has the highest FPR. Without the projection those three disagree completely, the typical key firing on no clean cell at all while the 95th-percentile key fires on 38 to 44% of them. Removing q=100 directions closes that spread onto the nominal line and lifts TPR at α=5% from 70.2 to 91.7 at n-gram=1. A small q does neither, because the leading directions carry watermark signal as well as shared task structure: the projection pays for itself only from $q\approx$10, after which power climbs monotonically. We deploy q=100, and App. C.6 reports the per-key calibration that remains at that setting.

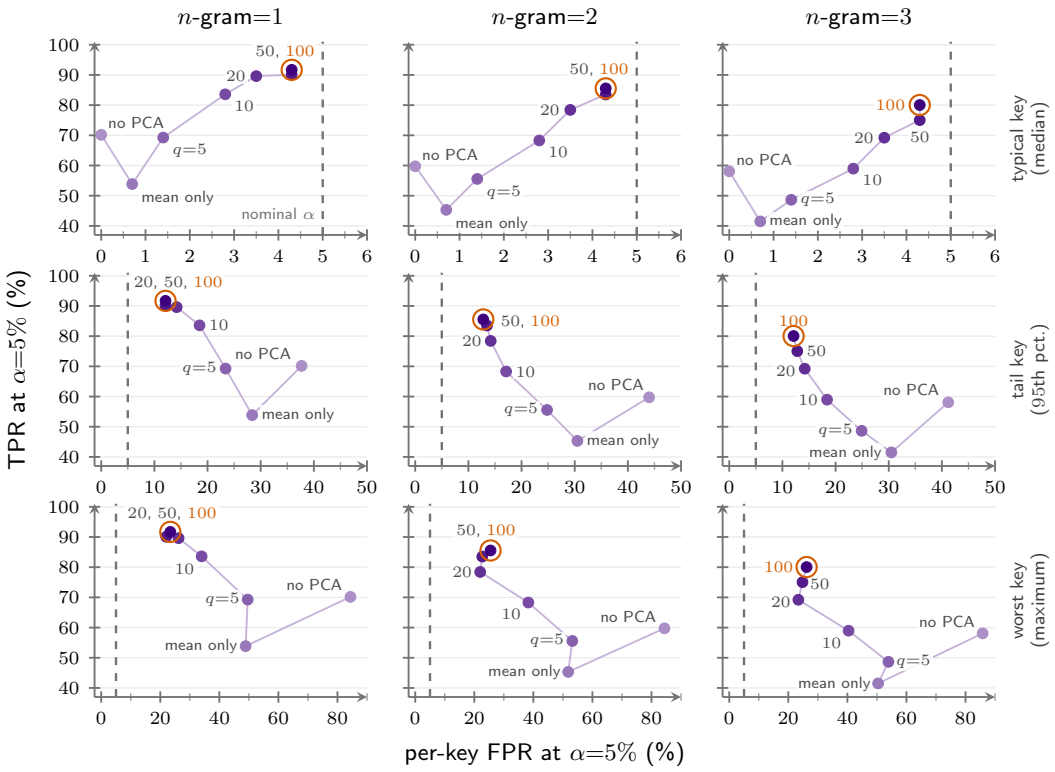


Figure 6: **Choosing q: detection power against three order statistics of the key bank.** The projection sweeps from *no PCA* through *mean only* (subtract the pool mean, remove no subspace) to q=100 removed components; the path visits the seven settings in that order and marker shade runs light to dark with q. **Columns** are the n-gram window. **Rows** are which key of the 1,000-key bank the false-positive rate is read off: the typical key, the 95th-percentile tail key and the worst key. y**:** TPR at α=5% over the 1,200 distilled students. x**:** the percentage of the 129 clean model-temperature cells on which that key fires at the same threshold.

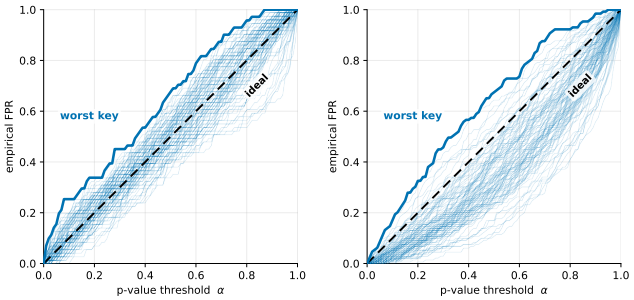
C.6 Calibration of the Rank p-Value

By the exchangeability argument of App. B.1, thresholding at $p \leq \alpha$ should falsely accuse an α fraction of models that were not distilled from the protected teacher, which is what §3.1 relies on when it reads operating points off the nominal p-value. We test it on 100 deployed keys and 65 clean model-temperature cells held out from the 64 used to fit the task subspace (App. C.5), ranking each key among the full 1,000-key bank. Figure 7 plots the empirical rate against α for each deployed key, for WM-Basin (Fig. 7a) and for the black-box baseline (Fig. 7b).

The per-key false-positive distribution. Table 11 reports the empirical false-positive rate at α=5% over the 1,000 keys of the bank, each read across the 129 clean model-temperature cells, after the task-subspace projection and under the same 4-fold cross-fitting by base model as App. C.5. The typical key is calibrated and the tail is not, which is the distinction the mean over the bank hides: the median key sits at 4.3% against a nominal 5% in all three windows and the 90th percentile at 9.9%, while the worst key reaches 23.4 to 26.2%. That tail is bounded by the size of the pool: the *floor* column is the level a perfectly exchangeable bank of this size reaches on its own (App. C.5), and against it the worst observed key is high by the *worst/floor* factor, where the unprojected scores are high by ≈6.

Table 11: **Per-key false-positive rate at α=5%, after removing the task subspace.** Each of the 1,000 bank keys is read over the 129 clean model-temperature cells, with 4-fold projectors cross-fitted by base model so no cell is scored by a subspace fit on its own model. All values are percentages except the last column. *Floor* is the 95th percentile of the maximum under exact uniformity at K=1,000 over 129 cells: the level a perfectly exchangeable bank of this size would still reach, and the reference the worst key should be judged against.

n-gram	median	p90	p95	worst	floor	worst / floor
1	4.3	9.9	12.1	23.4	13.5	1.74
2	4.3	9.9	12.8	25.5	13.5	1.89
3	4.3	9.9	12.1	26.2	13.5	1.95



(a) WM-Basin (ours) (b) Black-box baseline

Figure 7: **Nominal p-value thresholds match empirical false-positive rates.** Each faint curve is one of the 100 deployed keys, reporting its empirical FPR over the 65 held-out clean model-temperature cells at n-gram=3 as the threshold α varies; the solid curve is the worst key and the dashed line is the calibrated target FPR=α.

D Additional Detection Results

This section follows the sweep order of §3.2, adding the windows, operating points and factors the main tables leave out, together with two studies the main text states only in passing: the suspect-text budget behind the fragility claim of §1, and the choice of probe model, together with the warm-up it is given, named in the detector setup of §3.1. The configuration of App. C is the reference point throughout, and each row of Tab. 9 names what it varies.

D.1 Operating Points and Distillation Temperature

Figure 8 sweeps the detection threshold continuously over the same T=1.0 distillation arm that Tab. 2 of §3.2 reads at three fixed thresholds, on an empirical false-positive axis rather than that table's nominal p-values, and Tab. 12 reports the same temperature sweep at all three n-gram windows, of which the main body carries the n-gram=3 block. Both detectors are scored on one shared set of students and innocent cells here, all 100 keys rather than the per-window cohorts the tables report, so the comparison to make is between the two curves rather than between a curve and a cell of Tab. 2: WM-Basin sits at or above the black-box baseline throughout the plotted range at every window, and strictly above it everywhere below the false-positive rate at which both saturate. The baseline's curve is the read that preceded the token matching both tables now use; that matching lifts eight of the nine black-box cells of Tab. 12 at 5% FPR, by up to 18 points, so the separation plotted here is wider than the matched one.

D.2 Cross-Teacher Detection

Section 3.2 reports cross-teacher detection in Tab. 3, whose three thresholds are empirical false-positive rates. The sampled-token baseline can only accumulate evidence at positions whose context window recurs, and student diversity orders it: gpt-oss students repeat only 65.3% of their 2-word

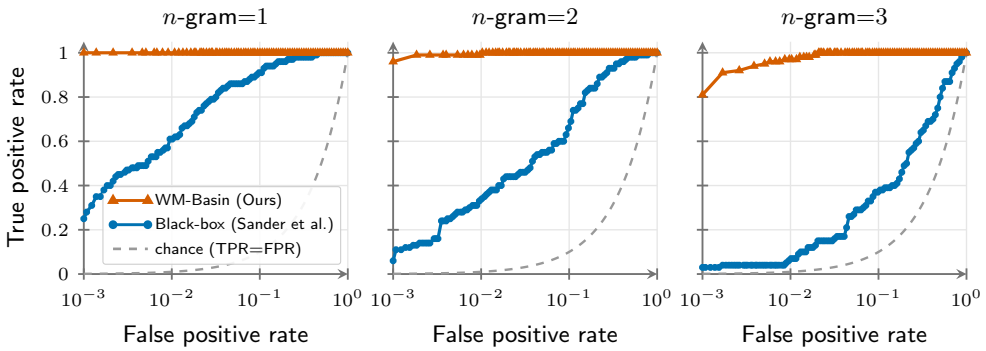


Figure 8: **Detection performance of the two closed-model detectors across all operating points.** ROC (log x, linear y), one panel per n-gram window, matched on the 100 distillation keys 1001 to 1100. **TPR** is over the 100 watermarked students (one per key, scored with its own key); **FPR** is over the same 100 keys $\times$ the 65 held-out innocent model-temperature cells (the evaluation half of the 129-cell clean pool: 43 off-the-shelf base models $\times$ 3 sampling temperatures; 6,500 null cases). Sweeping the threshold α traces each curve; WM-Basin (ours) is drawn in vermilion and Black-box (baseline) in blue. The grey dashed line marks chance (TPR=FPR).

Table 12: **Distillation detection across distillation sampling temperatures, all three n-gram windows.** The suspect model (Qwen2.5-3B) is distilled from the teacher at each distillation temperature, while the detector's reference bank stays fixed at T=1.0. Each cell is the percentage of up to 100 distilled models at that temperature, one per watermark key. The n-gram=3 block is Tab. 2 in the main body.

Temp.	n-gram=1						n-gram=2						n-gram=3					
	WM-Basin			Black-box			WM-Basin			Black-box			WM-Basin			Black-box		
	TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR		
	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%
1.0	100	100	100	72	43	26	100	100	98	50	25	9	100	99	88	23	9	1
0.7	100	100	100	37	16	2	100	97	88	25	12	0	99	93	62	12	2	0
0.5	100	100	99	24	7	2	94	78	40	8	2	1	71	51	22	4	2	0

contexts and the baseline reaches 18% there, against 92.1% and 93% on the most repetitive Llama students. Neither tokenizer sharing, trace length nor utility separates the teachers the same way (Tab. 13), and gpt-oss gives the second-largest GSM8K gain alongside the weakest sampled-token result. WM-Basin aggregates the watermark's gradient direction over all positions instead, and stays at 96 to 100% in the 5% column on every teacher.

Table 13: **Teacher properties, student utility and detection, for the three swapped-in teachers.** *DS tok* is the DeepSeek-R1-Distill tokenizer. Accuracy is mean ± standard deviation over 100 keyed Qwen2.5-3B students, and Δ is against the un-distilled base (GSM8K 48.5%, Math-500 33.8%). The two detection rows are the FPR=5% column of Tab. 3.

	gpt-oss-120B	QwQ-32B	Llama-3.1-8B
Teacher			
Family / tokenizer	MoE / 200k native	Qwen2.5 / DS tok	Llama / native
Median trace chars	10,552	23,095	2,969
Student utility			
GSM8K (%)	77.2 ± 1.7	80.8 ± 2.7	74.9 ± 1.2
Δ GSM8K	+28.7	+32.3	+26.4
Math-500 (%)	47.9 ± 1.5	52.1 ± 1.5	36.5 ± 1.5
Δ Math-500	+14.1	+18.3	+2.7
Detection, TPR at the 5% threshold			
WM-Basin	96	99	100
Black-box	18	88	93

D.3 Watermark Scheme

Table 4 replaces the teacher's TextSeal decoder with three other watermarking schemes. TextSeal (Sander et al., 2026), the dual-key scheme used everywhere else in the paper, draws each token's Gumbel noise from one of two keys with probability α=0.1; GumbelMax (Aaronson, 2022) is the single-key Gumbel scheme; and SynthID (Dathathri et al., 2024) resolves each position through a tournament of depth 10 between keyed g-values. GreenList (Kirchenbauer et al., 2023) instead adds a fixed logit bias δ=2.0 to a keyed half of the vocabulary (γ=0.5) at every position, the one scheme here that modifies the teacher's logits rather than the sampling rule. Each is generated at n-gram=1, 2 and 3 and distilled into 50 students, with all detectors reading the same ≈1M-token budget, the reference bank rebuilt per scheme and window, and everything else at the defaults of Tab. 8.

Does WM-Basin's margin hold when the baseline is read in its strongest form? The black-box statistic can weight each scored token by the teacher's entropy at that position, concentrating the test on the high-entropy positions where the watermark is able to bias the sample at all, as in (Sander et al., 2026); Tab. 14 gives the baseline that weighted read as a second row per scheme, beside the plain statistic reported in Tab. 4. The weighting is worth 8 to 30 points to the baseline at $p \leq 0.05$ across all twelve scheme and window cells, yet WM-Basin still leads at every level in every cell where that level is defined. Its margin in the main body is therefore not an artefact of reading the baseline in its plain form.

Table 14: **Distillation detection across watermarking schemes, with the baseline's entropy-weighted statistic.** The full form of Tab. 4: TextSeal, the teacher's default scheme, joins the three swapped-in ones, and each watermark occupies two rows, the black-box baseline (Sander et al., 2024) reading the suspect text with its plain statistic on the first and with its entropy-weighted statistic on the second. The weighting belongs to that statistic, so one WM-Basin run serves both rows and its cells span the pair. Cells are read as in Tab. 4, and the dash marks a level the 991-key TextSeal bank at n-gram=1 cannot resolve.

Watermark	n-gram=1						n-gram=2						n-gram=3					
	WM-Basin			Black-box			WM-Basin			Black-box			WM-Basin			Black-box		
	TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR			TPR@FPR		
	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%
TextSeal	100	100	—	76	44	28	100	100	98	52	30	6	100	98	84	22	8	2
TextSeal with entropy				96	88	60				74	54	40				46	24	2
GumbelMax	100	100	100	72	44	28	100	98	98	52	32	6	100	100	100	22	10	0
GumbelMax with entropy				96	88	62				80	54	28				46	18	6
GreenList	100	100	100	58	26	8	94	86	56	40	20	4	90	68	34	16	2	2
GreenList with entropy				66	46	28				54	32	8				38	10	0
SynthID	100	100	100	64	38	8	100	100	98	44	16	0	100	96	78	24	8	4
SynthID with entropy				94	68	38				66	38	12				42	22	6

D.4 Audited Domain and System Prompt

The audited-domain sweep of §3.2 reads Tab. 5; this section gives the per-domain rebuild and the exact prompts behind it. Every row only uses the prompt from the audit domain: the probe warm-up corpus, the PCA task subspace removal, the decoy bank, and the black-box baseline's per-key corpus filter are all built from that row's own questions and system prompt, never the teacher's math-teacher prompt, so neither detector inherits an advantage from the distillation domain.

Each suspect is ranked against its own key plus the 999 decoys of that row's bank, and 100 students are scored in every cell. All five rows read that row's own held-out test pool, the three math-dataset rows under the shared `math_neutral` system prompt and the other two under their own, so wording is constant inside the top block and the dataset alone separates those rows. On OpenR1-Math, the distilled dataset itself with held-out questions read under a prompt no suspect trained under, WM-Basin reaches 95%/83%/45% when context window is 1/2/3, so a new system prompt does not by itself cost the verifier its read.

Training domain (math): what the students actually learned on

```
You are a math teacher.  You will be given a math problem and you will
solve it step by step.
You will output your final solution like \boxed{ANSWER}.  Be sure
to include relevant units within the brackets and fully evaluate
arithmetic expressions.
```

math_neutral (OOD, closest to training)

```
You are a helpful assistant.  Solve the following problem step by step,
then give the final answer.
```

code (OOD)

```
You are an expert programmer.  Read the problem and write a complete,
correct Python solution.  Think step by step, then provide the final
code.
```

chat (OOD, farthest from training)

```
You are a helpful, honest, and courteous assistant.  Follow the user's
instruction and give a clear, complete response.
```

D.5 Suspect-Text Budget

How much suspect text does a verdict need, and does reading more always help? Section 3.1 fixes the default budget the in-domain tables are read at. We read the same fixed pool of 100 students, at cumulative prefixes of their generations, from 5 to 1,878 generations (0.03 to 10.3M tokens supplied), holding the probe, bank, task subspace and threshold fixed, and score both detectors on every prefix at 1% FPR (Fig. 9). WM-Basin accumulates evidence monotonically at every window, so more text never costs the verifier anything and no stopping rule is needed: it reaches 100% TPR by 0.6M tokens at n-gram 1 and 2, and at n-gram=3 it is within 8 points of its ceiling by that budget and spends the twenty times more text that follows gaining the rest. The baseline is not monotone in the budget at all: it peaks within the first ≈0.5M tokens and decays afterwards, because its scored-token denominator keeps growing while deduplication caps the signal it can collect.

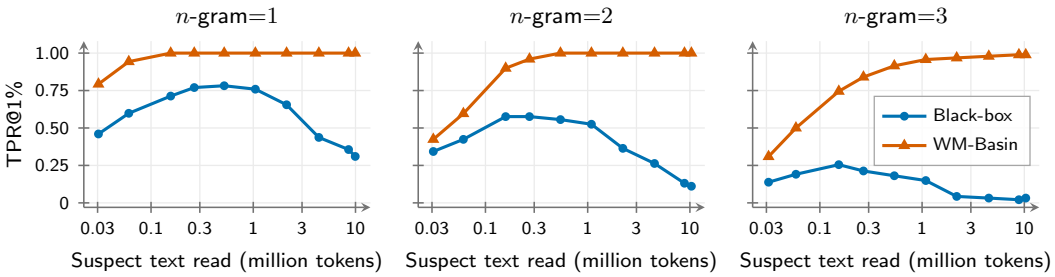


Figure 9: **Impact of suspect text budget on detection performance.** We plot True Positive Rate at 1% FPR (TPR@1%; share of distilled models detected at rank-$p \leq 0.01$, one model per key) against the amount of suspect text evaluated ($\log x$ scale, million tokens). Both WM-Basin (ours) and **Black-box** (Sander et al., 2024) are evaluated on identical generations at the same 1% FPR operating point.

D.6 Probe Model and Warm-Up

The probe model turns text into a gradient signal. In this section, we study the impact of the probe model on the performance of WM-Basin.

Choice of probe model. Does the verifier need a probe model from the suspect's own family? Table 15 varies only the probe θ_{probe} that §3.1 holds at a clean Qwen2.5-3B base, across ten checkpoints from 0.5B to 8B spanning four Qwen2.5 scales or post-training states, three Llama-3 checkpoints, and two families whose tokenizers sit farthest from the student's 152k vocabulary. Every probe receives the same warm-up, and because each defines a different gradient coordinate space its reference bank is rebuilt with the default recipe; the suspect cohort, clean PCA pool and three n-gram windows stay fixed. Family costs the detector nothing: every probe recovers all suspects at n-gram=1, and at n-gram=3 and 0.1% FPR the three Llama-3 probes land inside or just under the band the Qwen probes span, despite sharing no tokenizer with the student. The loss instead tracks how far the probe's vocabulary sits from the student's, the smallest vocabulary losing most while still staying an order of magnitude above the baseline in the same cell, and falling further behind the Qwen probes as the threshold tightens at n-gram=2 and 3. The verifier can therefore audit with whatever checkpoint it holds, preferring a nearby vocabulary.

Impact of warm-up. The probe warm-up of §3.1 is one short LoRA pass on clean teacher traces, run before any key gradient is taken; Tab. 16 sweeps it one knob at a time from the default, in a single scoring pass per arm. We report n-gram=3 because it is the only window that separates the arms: at n-gram=1 every probe in Tab. 15 already reads 100 at all three levels.

Removing the warm-up costs most of the detector's power at 0.1% FPR: an untrained, randomly initialised adapter keeps about a third of the default's TPR there, and the bare base model almost none. Gradients are taken with respect to the adapter parameters ϕ (§2.3), so a random adapter still defines a usable coordinate space and retains real signal at 5% FPR, whereas the bare model has no ϕ to differentiate against. The warm-up therefore sharpens a signal the probe already carries rather than creating one.

Table 15: **Effect of the probe model on detection.** The same 100 distilled suspects are held fixed while the verifier changes the probe model and rebuilds the reference bank. Probes vary in scale, post-training, family, and tokenizer; the **Black-box** baseline (Sander et al., 2024) uses no probe and is *token-matched*, reading exactly the same suspect text as WM-Basin. Each cell reports True Positive Rate (TPR, %) at FPR ($5\%, 1\%, 0.1\%$), over the suspects whose generations both detectors can score, $87/95/90$ of the 100 keys at n-gram $1/2/3$. At n-gram=3 and 0.1% FPR the five Qwen2.5 probes span 69 to 89 and the three Llama-3 probes 67 to 72; vocabulary size is given beside each cross-family row.

Probe model θ_{probe}	n-gram=1			n-gram=2			n-gram=3		
	TPR@FPR			TPR@FPR			TPR@FPR		
	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%
Black-box baseline (no θ_{probe})	72	34	11	44	28	9	26	13	3
Qwen2.5-3B-Base (default)	100	100	100	100	100	98	100	99	89
Qwen2.5-1.5B-Base	100	100	100	100	100	99	97	90	72
Qwen2.5-0.5B-Base	100	100	100	100	99	97	99	96	81
Qwen2.5-3B-Instruct	100	100	100	100	100	98	100	99	84
Qwen2.5-7B-Instruct	100	100	100	100	100	96	96	87	69
Llama-3.1-8B-Instruct (128k vocab)	100	100	100	100	100	92	100	92	67
Llama-Guard-3-8B (128k vocab)	100	100	100	100	100	94	97	91	69
Llama-3.2-1B-Instruct (128k vocab)	100	100	100	100	100	93	99	88	72
Gemma-2-2B-Base (256k vocab)	100	100	100	100	99	91	91	74	57
SmolLM2-1.7B-Base (49k vocab)	100	100	100	100	97	71	90	68	34

Warming on prompts from different domains and datasets, i.e., chat, MMLU, GSM8K or MATH instead of the audited OpenR1-Math distribution costs 11 to 21 points at 0.1% FPR and at most one key at 5%. Code is the single exception (37%), and it is the warm-up corpus farthest from the reasoning traces the reference bank is built from. This is a separate question from where the verifier draws its *detection* prompts, which do have to come from the audited domain or a similar one (§3.2).

Adapter rank and learning rate each move TPR at 0.1% FPR by at most 4 points, and epoch count by 11; none of the three moves the 5% column at all. Only divergence is fatal: at learning rate $1\text{e}{-3}$ the warm-up run diverges and detection collapses at every level, a training failure the verifier can see without running the detector at all. The cheapest arms we ran are among the best at every level, so the setting to deploy is whatever converges: rank 32 matches the default at 0.1% FPR and beats it at 1%, and one epoch beats it at 0.1%. We did not run the two together, so rank 32 at one epoch is an inference from two single-knob arms rather than a measured configuration.

E Additional Robustness Results

This section gives the full protocol and results omitted in §3.3, followed by a robustness analysis against paraphrasing attack: rewriting the teacher's traces before distillation. Each modifies the student or the traces it was distilled on and re-reads it with the same probe and reference bank, at the defaults of App. C unless this section or a caption records otherwise.

E.1 Compression

Section 3.3 reports compression at n-gram=3; this section adds the two shorter windows and the displacement of every operation. We apply per-output-channel symmetric int8, uniform group-64 int4, nf4, or 10 to 30% unstructured magnitude pruning to 100 keyed students at each n-gram window, regenerating detector inputs after each operation. Quantized weights are dequantized to fp16 for evaluation, and pruning excludes `lm_head` and `embed_tokens`. A model counts as usable when its mean GSM8K accuracy is at least $0.9\times$ the distilled baseline, which int8 and prune10 clear and int4, nf4, prune20 and prune30 do not. The *baseline* row is the uncompressed reference rather than a compression condition, so its black-box cells were never re-read under the token matching the six

Table 16: **The probe warm-up is necessary, but almost none of its details are.** All rows are WM-Basin at n-gram=3 on the same 90 watermark keys, the same suspects, and the same suspect-text budget; only the probe warm-up changes, one knob at a time from the default. Cells are TPR (%) at 5%, 1%, and 0.1% FPR, scored on the task-neutral *residual* gradients of §2.3. Defaults in Tab. 8.

Probe warm-up	WM-Basin			Probe warm-up	WM-Basin		
	5%	1%	.1%		5%	1%	.1%
(a) Warm-up presence				*(c) Adapter rank*			
Trained warm-up (default)	100	99	89	Rank 32	100	100	89
Untrained random adapter	82	59	33	Rank 128 (default)	100	99	89
No adapter (bare base model)	30	21	6	Rank 512	100	100	86
(b) Warm-up corpus				*(d) Learning rate*			
OpenR1-Math (default)	100	99	89	5e−5	100	98	90
MATH	100	97	76	2e−4 (default)	100	99	89
GSM8K	99	96	68	5e−4	100	100	93
MMLU	99	97	69	1e−3 (diverged)	7	2	0
Chat	99	94	78	*(e) Warm-up epochs*			
Code	88	73	37	1 epoch	100	99	92
				4 epochs (default)	100	99	89
				8 epochs	100	97	81

Table 17: **Distillation detection under compression, all three windows.** The two windows omitted from Tab. 6, with n-gram=3 repeated for reference; rows and cohort (100 suspects per cell) are as there, and cells are read as in Tab. 2, except that the *baseline* row's black-box cells are not token-matched to WM-Basin's read and are therefore not comparable down the black-box columns. `Acc.` is the per-window mean GSM8K, and operations below the mid-rule are unusable, at Acc. under $0.9\times70.7=63.6$. The weight-space decomposition for every row is in Tab. 18.

Compression	n-gram=1							n-gram=2							n-gram=3						
	Acc.	WM-Basin TPR@FPR			Black-box TPR@FPR			Acc.	WM-Basin TPR@FPR			Black-box TPR@FPR			Acc.	WM-Basin TPR@FPR			Black-box TPR@FPR		
		5%	1%	.1%	5%	1%	.1%		5%	1%	.1%	5%	1%	.1%		5%	1%	.1%	5%	1%	.1%
baseline	70.7	100	100	100	81	50	17	70.7	100	100	99	44	19	5	70.7	100	95	80	8	4	2
int8	72.2	100	100	100	75	50	17	71.0	100	100	99	50	27	6	70.0	100	99	85	24	7	1
prune10	77.1	100	100	100	76	49	17	76.2	100	100	98	52	27	9	77.3	99	94	77	21	8	1
prune20	59.5	100	100	100	70	44	20	61.2	100	96	82	44	24	10	57.6	91	82	49	20	4	0
int4	50.4	79	59	31	50	25	11	50.8	43	25	12	38	16	2	45.4	46	25	10	7	1	0
nf4	55.8	100	100	99	70	46	23	42.9	100	95	76	48	26	4	41.2	95	79	40	8	3	0
prune30	37.8	93	71	36	52	30	8	39.6	75	51	12	32	13	1	28.9	49	26	7	14	1	0

operation rows use; in Tab. 6 and Tab. 17 alike, that row therefore supports no black-box comparison, either with its own WM-Basin cells or down the column.

Compression costs the watermark only where it costs the model. Int8 and prune10 retain both utility and strong WM-Basin detection, and the larger detection losses occur only beyond the usability gate, with nf4 and prune20 the two operations where the gradient signal outlasts utility (Fig. 10 at n-gram=3; Tabs. 17 and 18 at all three windows). The weight-space displacement is ordered the same way: the along component of every compression operation stays below 0.11 against orthogonal movement of 14 to 152, and grows with the pruning fraction. Therefore, compression does not move the model outside the watermark basin, and detection remains effective.

E.2 Continued Fine-Tuning and RL

Section 3.3 reports continued SFT and RL at n-gram=3. This section adds the two shorter windows, the training recipes, and the displacement of every arm. We continue training the keyed Qwen2.5-3B

Table 18: **Compression moves almost entirely orthogonally to the watermark direction.** Across 100 students per window, Acc. is mean GSM8K (%) and *Usable?* denotes Acc.$\geq 0.9\times 70.7{=}63.6$. We decompose $\Delta_{\text{post}}{=}\boldsymbol{\theta}_{\text{post}}{-}\boldsymbol{\theta}_{\text{wm}}$ into its signed projection along the watermark key direction $\hat{\Delta}_{\text{wm}}$ (Eq. (1)) and the orthogonal L_2 remainder. The remainder is a property of the compressed weights alone and is identical at the three windows; the usability verdict is read off each window's own Acc. and comes out the same in every row. Both are therefore printed once. The along component is at least three orders of magnitude smaller and increases in magnitude within the ordered pruning sweep.

			n-gram=1		n-gram=2		n-gram=3	
Cond.	Orthogonal	Usable?	Acc.	Along w	Acc.	Along w	Acc.	Along w
baseline	0	✓	70.7	0	70.7	0	70.7	0
int8	14.09	✓	72.2	−0.0001	71.0	−0.0002	70.0	−0.0001
int4	152.07	✗	50.4	−0.0113	50.8	−0.0112	45.4	−0.0117
nf4	125.91	✗	55.8	−0.0291	42.9	−0.0281	41.2	−0.0266
prune10	27.23	✓	77.1	−0.0063	76.2	−0.0061	77.3	−0.0058
prune20	77.74	✗	59.5	−0.0390	61.2	−0.0368	57.6	−0.0348
prune30	144.85	✗	37.8	−0.1022	39.6	−0.0945	28.9	−0.0887

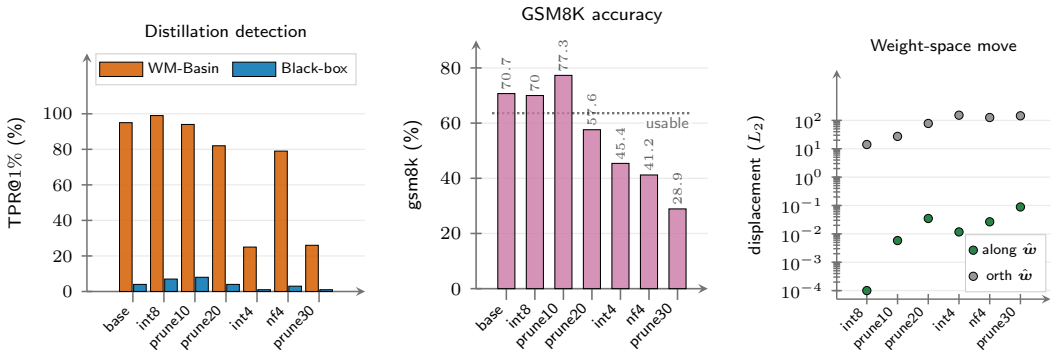


Figure 10: **Robustness to post-processing.** TextSeal, n-gram=3, 100 keyed students under int8, int4 and nf4 quantization and $10/20/30\%$ pruning; *base* is uncompressed. **(a)** TPR at 1% FPR for WM-Basin and Black-box; **(b)** gsm8k accuracy, dotted line is the $0.9\times$ usable gate; **(c)** weight-space move along the watermark direction vs orthogonal residual. Details in Tab. 18.

students of each n-gram window on clean data. We then regenerate suspect text and re-measure detection, utility and geometry. Continued SFT uses one of two clean corpora. The *reason* corpus is 1,900 traces, 950 code and 950 science, from open-r1/Mixture-of-Thoughts (Hugging Face, 2025). The *multilingual* corpus is 1,206 traces on math questions in eight languages (Chen et al., 2024). Both pass the think-closure and repetition filters of App. C.2.

SFT keeps the distillation adapter shape. It runs 2 epochs at global batch 16 and maximum sequence length 8192, on a cosine learning rate of 2×10^{-5} with 0.05 warmup. Continued RL uses GRPO (Shao et al., 2024). The in-domain arm optimises gsm8k with a verifiable correctness reward. The two cross-domain arms use learning rate 5×10^{-6} on rule-verified K&K logic puzzles (Xie et al., 2025) or on RLVR-IFeval instruction following (Lambert et al., 2024). Both cross-domain rewards are all-or-nothing: K&K requires every inhabitant to be labeled correctly in the final boxed answer, and IFEval requires the requested constraint to hold. We measure task accuracy greedily on 400 held-out puzzles and 1,000 held-out prompts. Detector inputs keep the standard $T{=}1.0$, top-$p{=}0.95$ sampling.

For every condition we decompose $\Delta_{\text{post}}{=}\boldsymbol{\theta}_{\text{post}}{-}\boldsymbol{\theta}_{\text{wm}}$ against $\hat{\Delta}_{\text{wm}}$, the key-specific watermark direction of Eq. (1). The Open-weight column (Sander et al., 2024) is teacher-forced on the adversary's own distillation traces. That is the ideal case, and not a read our verifier can run. It shows that those traces stay memorised after continued training, not that the watermark survives in the weights.

Table 19: **Distillation detection under continued training, the two remaining windows.** The windows omitted from Tab. 7; rows, attacks, and protocol are as there, and cells are read as in Tab. 2. Cohorts differ by window: each SFT row pools 100 suspects and each GRPO row pools 10 suspects. Every detector column of a row shares that row's denominator. Δ`Acc.` is the paired per-key change in GSM8K, and the weight-space geometry for both windows is in Tab. 20.

	n-gram=1										n-gram=2									
Cond.	ΔAcc.	WM-Basin			Black-box			Open-weight			ΔAcc.	WM-Basin			Black-box			Open-weight		
		TPR@FPR			TPR@FPR			TPR@FPR				TPR@FPR			TPR@FPR			TPR@FPR		
		5%	1%	.1%	5%	1%	.1%	5%	1%	.1%		5%	1%	.1%	5%	1%	.1%	5%	1%	.1%
base	—	100	100	100	66	43	7	100	99	92	—	100	100	99	32	17	6	100	100	100
Continued SFT on clean cross-domain data																				
reason	+4.8	3	1	0	39	15	3	92	82	52	+1.0	4	2	0	22	9	1	99	97	76
multi.	+7.0	33	16	3	63	34	13	99	97	89	+4.6	27	13	3	36	18	5	100	100	99
Continued RL (GRPO) on in-domain math (gsm8k)																				
rl (gsm8k)	+2.4	100	100	100	58	42	25	100	100	92	+2.2	100	100	87	40	40	7	100	100	100
Continued RL (GRPO) on cross-domain verifiable tasks																				
rl (K&K)	+0.5	100	100	100	57	14	0	100	100	86	+0.9	100	100	100	40	30	20	100	100	100
rl (IFEval)	−1.4	100	100	100	71	57	14	100	100	86	+1.0	100	100	100	30	30	10	100	100	100

Table 20: **GRPO preserves utility with far smaller weight movement than SFT.** Every row is paired against its own cohort's pre-attack students, whose accuracy is printed as *base*: the cohorts differ in size (n) and do not share a baseline, so Δ is the mean per-key change and not the difference of the two pooled columns beside it. GSM8K and Math-500 are percentages and Δ is in points. Move (total) is $\|\Delta_{\text{post}}\|_2$, Along w is its signed projection on the key direction $\hat{\Delta}_{\text{wm}}$, and Along/total is the absolute percentage share. GRPO projections are under 0.0005 in magnitude and print as 0.000. The SFT rows share the full 100-key pool as their base; the GRPO rows are smaller cohorts carrying their own, which run from 0.6 points above it to 2.8 below, which is why Tab. 7 reports accuracy as a paired change rather than an absolute.

			GSM8K			Math-500			Move		
Cond.	n-gram	n	base	post	Δ	base	post	Δ	total	along w	along/total
reason	1	100	67.2	72.0	+4.77	41.9	40.8	−1.11	4.80	−0.252	5.3%
reason	2	100	69.0	70.0	+0.96	43.0	39.4	−3.60	4.70	−0.182	3.9%
reason	3	100	69.0	71.3	+2.37	43.2	40.6	−2.62	4.66	−0.146	3.1%
multilingual	1	100	67.2	74.2	+6.98	41.9	46.9	+4.99	2.72	−0.013	0.5%
multilingual	2	100	69.0	73.6	+4.59	43.0	46.7	+3.68	2.71	−0.009	0.3%
multilingual	3	100	69.0	75.0	+6.03	43.2	47.5	+4.29	2.70	−0.007	0.3%
rl (gsm8k, GRPO)	1	20	67.8	70.1	+2.35	41.3	42.2	+0.90	0.051	0.000	0.0%
rl (gsm8k, GRPO)	2	20	69.3	71.5	+2.17	42.5	43.2	+0.74	0.049	0.000	0.0%
rl (gsm8k, GRPO)	3	20	68.0	68.5	+0.52	43.4	44.0	+0.67	0.050	0.000	0.0%
rl (K&K, GRPO)	1	10	66.4	66.9	+0.48	40.9	42.4	+1.44	0.228	0.000	0.0%
rl (K&K, GRPO)	2	10	67.8	68.7	+0.93	42.9	43.4	+0.54	0.226	0.000	0.0%
rl (K&K, GRPO)	3	10	66.2	66.0	−0.20	43.7	44.8	+1.04	0.241	0.000	0.0%
rl (IFEval, GRPO)	1	10	66.4	65.0	−1.42	40.9	40.3	−0.58	0.242	0.000	0.0%
rl (IFEval, GRPO)	2	10	67.8	68.7	+0.96	42.9	42.1	−0.76	0.251	0.000	0.0%
rl (IFEval, GRPO)	3	10	66.3	66.1	−0.26	43.3	41.6	−1.71	0.246	0.000	0.0%

Which of these updates removes the watermark? One of the two SFT arms does, and no RL arm does (Tabs. 7 and 19). The *reason* arm puts WM-Basin at its wrong-key floor at all three windows. The *multilingual* arm only degrades it, and part of the cohort is still recovered at 5% FPR. WM-Basin stays flat against each GRPO cohort's pre-attack detection at 5% FPR, at every window. The Open-weight

Table 21: **Cross-domain GRPO improves the task it optimises.** Pre, post and Δ are task accuracy in percent on greedy held-out evaluation (400 K&K puzzles, 1,000 IFEval prompts). Each task and window pools 10 keys, except IFEval at n-gram=3 (11). Counted over the three windows together, 10 keys improved on K&K and 10 on IFEval. The same students' weight-space displacement and their off-target GSM8K and Math-500 changes are in Tab. 20.

	n-gram=1			n-gram=2			n-gram=3		
Task	Pre	Post	Δ	Pre	Post	Δ	Pre	Post	Δ
K&K	10.0	14.4	+4.4	9.0	13.4	+4.4	10.1	15.2	+5.1
IFEval	18.4	21.3	+2.9	18.3	21.6	+3.3	18.2	22.1	+3.9

column stays at 86 or above in 47 of its 51 cells, and its four exceptions are all in the *reason* arm. The same detector scored on held-out prompts falls by 41 to 87 points. Every update is nearly orthogonal to $\hat{\Delta}_{\mathrm{wm}}$, SFT and GRPO alike. Its projection on the watermark direction is at most 5.3% of its length under SFT and 0.0% under every GRPO arm (Tab. 20). The watermark component of the weights is left in place, and under GRPO so is the text read. Continued SFT changes what the student writes instead. WM-Basin and the black-box baseline score that text alone, so both fall on the *reason* arm at 5% FPR.

E.3 Paraphrasing

We paraphrase only the reasoning body between `<think>` and `</think>`, preserving the tags and the final answer, then reapply the standard quality filter before distillation. A same-family (Qwen) paraphraser rewrites the traces under mild, moderate and aggressive instructions at T=0.5, 0.8 and 1.0 respectively, with top-p=0.95 and at most 8,192 generated tokens. Each setting trains 20 keyed students. Rewriting costs the adversary no capability: mean GSM8K accuracy exceeds 77% at every strength and window against 68 to 69% for the unmodified control, and every strength stays above its matched clean student in the paired per-key comparison.

Each n-gram window is a block of Tab. 22. Mild rewriting leaves the watermark readable from text: WM-Basin holds 100, 100 and 90% TPR at 5% FPR at n-gram=1, 2 and 3, against 47, 20 and 10% for the black-box baseline. Moderate and aggressive rewriting bring both text detectors down together, to at most 33% TPR at 5% FPR for WM-Basin and 15% for the baseline. What is left then depends on which traces the open-weight read (Sander et al., 2024) is teacher-forced on. On the original teacher traces, the set a verifier actually holds, it falls with the text detectors (53, 55 and 55% at 5% FPR under moderate rewriting). On the rewritten traces the student trained on, it recovers the moderate arm almost in full (87, 100 and 100% at 5% FPR, and 95% at 1% FPR at the two longer windows). The aggressive arm is gone under either read, at best 33 to 55% at 5% FPR. Moderate rewriting therefore leaves the watermark in the weights and only hides it from the traces the verifier can obtain.

Table 22: **Effect of paraphrasing teacher traces before distillation, at all three n-gram windows.** Every watermarked trace is rewritten at one of three strengths with a same-family (Qwen) paraphraser; *none* is the unmodified control, repeated in each block because each window is a separate training set. WM-Basin (ours) and **Black-box** (Sander et al., 2024) score text, while **Open-weight** (Sander et al., 2024) teacher-forces the original teacher traces (*orig.*) or the rewritten traces the student actually trained on (*ideal*), the latter an upper bound that assumes the verifier already holds the adversary's rewritten training set. Each cell reports TPR (%) at FPR ($5\%, 1\%, 0.1\%$) over 20 students, except at n-gram=1, which is scored over the 15 of 20 keys whose reference direction at that window is clean and whose strictest column is 0.101% FPR, the finest rate that bank resolves.

Strength	WM-Basin			Black-box			Open-weight					
	TPR@FPR			TPR@FPR			orig.			ideal		
	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%	5%	1%	.1%
n-gram=1 (15 of 20 keys; strictest column is 0.101% FPR)												
none	100	100	100	60	20	0	100	93	93	100	93	93
mild	100	100	100	47	13	0	93	87	60	100	100	100
moderate	33	0	0	7	0	0	53	13	0	87	73	20
aggressive	13	0	0	7	0	0	7	7	0	33	13	7
n-gram=2												
none	100	100	100	60	25	5	100	100	100	100	100	100
mild	100	85	65	20	10	5	100	100	95	100	100	100
moderate	15	15	5	15	5	5	55	20	0	100	95	85
aggressive	15	10	5	5	5	0	5	5	0	55	40	10
n-gram=3												
none	100	100	85	35	10	0	100	100	100	100	100	100
mild	90	70	25	10	5	0	100	100	95	100	100	100
moderate	5	0	0	5	0	0	55	25	15	100	95	80
aggressive	0	0	0	0	0	0	10	0	0	55	25	10