GRADIENT-GUIDED RADIOACTIVE WATERMARKING FOR DETECTING FINE-TUNING ON TEACHER-GENERATED DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Model distillation transfers capabilities from powerful large language models (LLMs) to smaller downstream models, raising an important provenance question: *Has a downstream model been fine-tuned on protected teacher-generated data?* Existing methods embed watermark into generated text to detect such data use. However, their watermark statistics are evaluated on training-related or same-prompt contexts. In real-world scenarios, model owners may not know which data is used for distillation. To address this gap, we introduce **Gradient-Guided Radioactive Watermarking** (GGRW), a generation-time watermarking framework that links token selection to downstream fine-tuning dynamics. GGRW uses a proxy model to identify and sample tokens whose induced fine-tuning updates reinforce a private watermark direction, enabling detection on disjoint held-out contexts used in neither watermark construction nor student fine-tuning. Experimental results show that the watermark is detectable under dilution attacks and distillation, where student architectures are unknown. Furthermore, our method effectively detects the watermark on disjoint held-out contexts, whereas the baseline methods fail.

1 INTRODUCTION

Large language models (LLMs) are increasingly used as teachers to train smaller downstream models, transfering model capabilities while reducing inference and deployment costs (Hsieh et al., 2023; Gu et al., 2024). However, this creates a provenance problem for model owners: it is difficult to determine whether a downstream model has been distilled from protected teacher outputs.

Text watermarking offers a potential approach to traceability. The red-and-green-list watermark (KGW) of Kirchenbauer et al. (2023) pseudorandomly partitions the vocabulary and increases the probability of green-list tokens during decoding. Although this technique was originally designed to identify machine-generated text, subsequent work has shown that such signals exhibit radioactivity: models fine-tuned on watermarked text may retain detectable traces of the watermark (Sander et al., 2024). This property has since been explored for benchmark-contamination detection (Sander et al., 2025). These results suggest that text watermarks can trace both generated content and downstream data use. However, in benchmark-watermarking settings, radioactivity is detected using benchmark-specific watermarked contexts. Moreover, the watermark is not explicitly optimized to propagate through downstream fine-tuning.

Recent research has begun to explore downstream learning dynamics. Antidistillation Fingerprinting (ADFP) (Xu et al., 2026) uses a proxy model to favor tokens that increase the green-list probability after fine-tuning, making watermark inheritance an explicit generation objective. However, the detection of ADFP relies on a same-prompt setting that reuses prompts from downstream fine-tuning. In real-world scenarios, model owners may not access the exact prompts or traces used for downstream fine-tuning, corresponding to disjoint held-out contexts (Sander et al., 2025). This raises broader questions: *Can we construct a watermark that survives downstream fine-tuning? Can watermark remain detectable on disjoint held-out contexts?*

To answer this question, we propose **Gradient-Guided Radioactive Watermarking (GGRW)**, a generation-time watermarking framework that explicitly optimizes watermark inheritance through

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

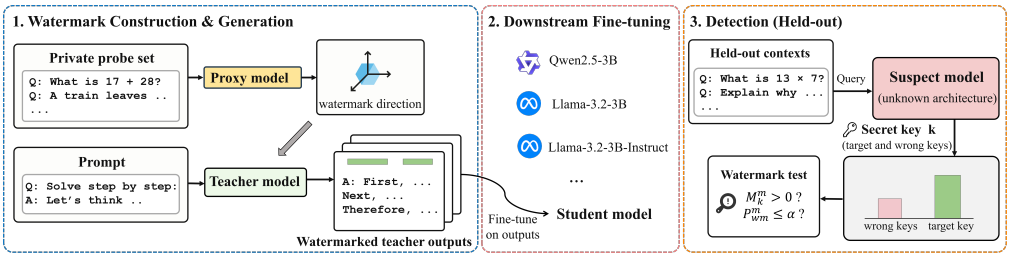


Figure 1: Overview of Gradient-Guided Radioactive Watermarking (GGRW).

downstream fine-tuning. Figure 1 illustrates the overall pipeline, including watermark construction and protected generation, downstream fine-tuning, and held-out detection. GGRW first defines a key-dependent watermark response as the target for downstream inheritance and uses a proxy model to derive the corresponding watermark direction. During generation, the teacher softly favors tokens whose induced fine-tuning updates align with this direction, thereby linking token selection directly to downstream watermark inheritance. Moreover, GGRW enables detection on disjoint held-out contexts that are excluded from both watermark construction and student fine-tuning.

Our main contributions are:

- We propose GGRW, a generation-time watermarking framework that uses proxy-model guidance to favor tokens whose induced fine-tuning updates reinforce a key-dependent watermark response.
- We develop a held-out detector that evaluates watermark inheritance on contexts used in neither watermark construction nor student fine-tuning.
- Experiments show that the GGRW transfers watermark signals across student architectures, remains detectable under substantial clean-data dilution, and can be detected on disjoint held-out contexts where the evaluated baselines remain undetected.

2 Related Work

LLM Watermarking and Radioactivity. Text watermarking has been widely studied for identifying LLM-generated content. Logit-based methods such as Kirchenbauer et al. (2023) and Zhao et al. (2023) bias next-token generation using keyed vocabulary partitions, while later work improves robustness, statistical guarantees, security, or distribution preservation (Fernandez et al., 2023; Kuditipudi et al., 2024; Christ et al., 2024; Liu et al., 2024b;a; Hu et al., 2024; Wu et al., 2024; Liu & Bu, 2024; Dathathri et al., 2024). These methods mainly detect the generated text itself. Beyond output-level detection, Sablayrolles et al. (2020) introduced the broader notion of radioactive data, and Sander et al. (2024) showed that language models trained on watermarked text can retain a detectable watermark signal. This property has subsequently been used for benchmark-contamination detection (Sander et al., 2025) and studied in unauthorized knowledge-distillation settings, including watermark-removal attacks (Pan et al., 2025; Yi et al., 2025). Our work instead optimizes token generation to produce a key-dependent watermark signal that persists through downstream fine-tuning.

Distillation Protection and Provenance. Knowledge distillation transfers behavior from a teacher to a student model (Hinton et al., 2015; Gou et al., 2021), and LLM distillation increasingly relies on teacher-generated supervision (Hsieh et al., 2023; Gu et al., 2024). Antidistillation sampling (Savani et al., 2025) uses a proxy model to modify teacher outputs so that the resulting traces are less useful for downstream distillation. Antidistillation Fingerprinting (ADFP) adapts this idea for provenance detection, using proxy-model gradients to favor tokens which increase the green-list probability of the fine-tuned student model (Xu et al., 2026). GGRW is most closely related to this line of work, but guides token selection according to the alignment between token-induced fine-tuning updates and the watermark direction. It further evaluates watermark inheritance on held-out contexts disjoint from watermark construction and student fine-tuning.

3 GRADIENT-GUIDED RADIOACTIVE WATERMARKING

3.1 PROBLEM SETTING

We consider a teacher model with parameters θ_T that releases generated outputs, a proxy model with parameters θ_P used to construct the watermark, and a downstream student with parameters θ_S that may later be fine-tuned on the teacher's outputs. Our goal is to modify teacher generation so that downstream fine-tuning induces a key-dependent watermark response that can later be detected.

Let $p(\cdot \mid c; \theta)$ denote the next-token distribution of a language model with parameters θ under context c. Given a private key k, we seek a protected teacher distribution that favors tokens satisfying two requirements. First, the selected tokens should remain plausible under the teacher's nominal distribution. Second, fine-tuning on these tokens should induce parameter updates that reinforce watermark response associated with k. Since the downstream student is unknown during watermark construction, we estimate this training effect using the proxy model.

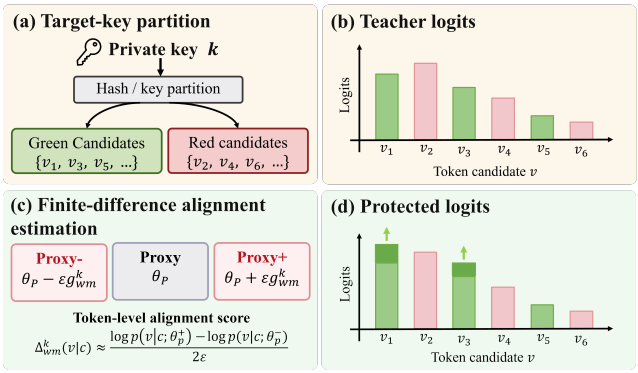


Figure 2: The protected generation procedure. (a) The private key partitions candidate tokens into target-key green and red sets. (b) The teacher provides the nominal logits over candidate tokens. (c) Token-level alignment with the watermark direction is estimated using two perturbed proxy models. (d) GGRW assigns a positive watermark reward only to target-key green candidates with positive alignment.

3.2 WATERMARK DIRECTION AND TOKEN-LEVEL ALIGNMENT

Key-dependent watermark response. We first define the behavior that we want downstream fine-tuning to reinforce. Let $\mathcal{E}_{\text{probe}} = \{(c_i, C_i)\}_{i=1}^{M}$ be a private probe set, where c_i is a context and C_i is a set of plausible next-token candidates defined by teacher. A private key k induces a pseudorandom partition of each candidate set into a green subset G_i^k and a red subset $R_i^k = C_i \setminus G_i^k$.

For any $S \subseteq C_i$, we define the probability mass assigned by a model with parameters θ as

$$P_\theta(S \mid c_i, C_i) = \frac{\sum_{v \in S} p(v \mid c_i; \theta)}{\sum_{u \in C_i} p(u \mid c_i; \theta)}. \tag{1}$$

We denote the teacher's nominal green probability mass by

$$\gamma_i^T = P_{\theta_T}(G_i^k \mid c_i, C_i). \tag{2}$$

We then define the key-dependent watermark response

$$R_k(\theta) = \frac{1}{M} \sum_{i=1}^{M} \left[\log \frac{P_\theta(G_i^k \mid c_i, C_i) + \xi}{\gamma_i^T + \xi} - \log \frac{P_\theta(R_i^k \mid c_i, C_i) + \xi}{1 - \gamma_i^T + \xi} \right], \tag{3}$$

where $\xi > 0$ is a small constant for numerical stability. Intuitively, $R_k(\theta)$ measures whether the model assigns greater probability mass to the target-key green candidates relative to the teacher's nominal green-red partition. The teacher normalization prevents an event from being strongly watermarked merely because the teacher already places substantial probability mass on its green subset.

Token-level training effect. R_k specifies the response that we seek to reinforce through downstream fine-tuning. Since generation operates at the token level, we next quantify how training on an individual candidate token changes this response.

Consider a candidate token v under context c. One gradient-descent step on its negative log-likelihood gives

$$\begin{aligned}\theta'_P(v) &= \theta_P - \eta \nabla_{\theta_P} \left[-\log p(v \mid c; \theta_P)\right] \\ &= \theta_P + \eta \nabla_{\theta_P} \log p(v \mid c; \theta_P),\end{aligned} \tag{4}$$

where $\eta > 0$ denotes the fine-tuning step size. The corresponding finite-step change in the watermark response is

$$\delta^{k,\eta}_{\mathrm{wm}}(v \mid c) = R_k(\theta'_P(v)) - R_k(\theta_P). \tag{5}$$

Its sign indicates whether the finite-step update increases or decreases the watermark response.

Directly evaluating Eq. (5) would perform a separate proxy update for every candidate token. We instead isolate the local training effect by considering the response per unit step in the limit $\eta \to 0$. Define

$$\begin{aligned}D^k_{\mathrm{wm}}(v \mid c) &\triangleq \lim_{\eta \to 0} \frac{\delta^{k,\eta}_{\mathrm{wm}}(v \mid c)}{\eta} \\ &= \lim_{\eta \to 0} \frac{R_k(\theta_P + \eta \nabla_{\theta_P} \log p(v \mid c; \theta_P)) - R_k(\theta_P)}{\eta} \\ &= \left\langle \nabla_{\theta_P} R_k(\theta_P), \nabla_{\theta_P} \log p(v \mid c; \theta_P) \right\rangle .\end{aligned} \tag{6}$$

Thus, $D^k_{\mathrm{wm}}(v \mid c)$ is the directional derivative of the watermark response along the parameter update induced by training on v. A positive value indicates that the update on v increases R_k. We use this local quantity as the approximate for the effect of downstream fine-tuning. The derivation follows from a first-order Taylor expansion and is provided in Appendix A.1.

Normalized watermark direction. The directional derivative in Eq. (6) depends on the raw scale of $\nabla_{\theta_P} R_k(\theta_P)$. In GGRW, this gradient is used primarily to define a direction in proxy-model parameter space. We normalize it as

$$g^k_{\mathrm{wm}} = \frac{\nabla_{\theta_P} R_k(\theta_P)}{\|\nabla_{\theta_P} R_k(\theta_P)\|_2}. \tag{7}$$

The corresponding normalized token-level alignment is

$$\begin{aligned}A^k_{\mathrm{wm}}(v \mid c) &\triangleq \left\langle g^k_{\mathrm{wm}}, \nabla_{\theta_P} \log p(v \mid c; \theta_P) \right\rangle \\ &= \frac{D^k_{\mathrm{wm}}(v \mid c)}{\|\nabla_{\theta_P} R_k(\theta_P)\|_2}.\end{aligned} \tag{8}$$

Because the normalization factor is positive and independent of v, A^k_{wm} preserves the sign and relative ordering of the original directional derivatives.

Efficient finite-difference approximation. Computing $A^k_{\mathrm{wm}}(v \mid c)$ directly for every vocabulary token would require token-specific gradients. We instead estimate it using a symmetric finite difference along the fixed watermark direction. Specifically, we construct two statically perturbed copies of the proxy model:

$$\theta^+_P = \theta_P + \epsilon g^k_{\mathrm{wm}}, \qquad \theta^-_P = \theta_P - \epsilon g^k_{\mathrm{wm}}, \tag{9}$$

where $\epsilon > 0$ is the finite-difference radius. For a token v under context c, we define

$$\Delta^k_{\mathrm{wm}}(v \mid c) = \frac{\log p(v \mid c; \theta^+_P) - \log p(v \mid c; \theta^-_P)}{2\epsilon}. \tag{10}$$

By applying a centered finite-difference approximation to the token log probability along g^k_{wm},

$$\begin{aligned}\Delta^k_{\mathrm{wm}}(v \mid c) &= A^k_{\mathrm{wm}}(v \mid c) + O(\epsilon^2) \\ &= \left\langle g^k_{\mathrm{wm}}, \nabla_{\theta_P} \log p(v \mid c; \theta_P) \right\rangle + O(\epsilon^2).\end{aligned} \tag{11}$$

A detailed derivation of the second-order approximation is given in Appendix A.2.

Hence, $\Delta^k_{\mathrm{wm}}(v \mid c)$ provides a practical estimate of the token-level alignment. Its sign indicates whether the token-induced update is locally aligned with the target watermark direction, while its magnitude measures the corresponding local alignment strength. The two perturbed proxy models are constructed once before generation; at each decoding step, two proxy-model forward passes provide $\Delta^k_{\mathrm{wm}}(v \mid c)$ for all vocabulary tokens simultaneously.

3.3 Gradient-Guided Protected Sampling

During protected generation, we use $\Delta^k_{\mathrm{wm}}(v \mid c_t)$ as the alignment score. At autoregressive step t, let $c_t = x_{1:t}$ denote the current context and $z_T(v \mid c_t)$ the teacher logit. We restrict intervention to a teacher-defined candidate set C_t, where the private key partitions candidates into G^k_t and R^k_t.

We use the positive part of the alignment score as the gradient-weighted key-dependent reward:

$$\rho^k_{\mathrm{wm}}(v \mid c_t) = \left[\Delta^k_{\mathrm{wm}}(v \mid c_t)\right]_+, \tag{12}$$

Let $\bar{z}_T(v \mid c_t) = \frac{z_T(v|c_t)}{\tau}$ be the temperature-scaled teacher logit. We define the watermark score as

$$s^k_{\mathrm{wm}}(v \mid c_t) = \mathbb{I}[v \in G^k_t]\rho^k_{\mathrm{wm}}(v \mid c_t) - \mu_{\mathrm{wm}}[\bar{z}_T(v_{\max} \mid c_t) - \bar{z}_T(v \mid c_t)], \tag{13}$$

where $v_{\max}$ is the teacher-preferred token and $\mu_{\mathrm{wm}} \geq 0$ controls the penalty for deviating from that token. The first term rewards target-key green candidates with positive alignment, while the second preserves teacher preference by penalizing candidates that deviate from the teacher-preferred token.

The protected logit is

$$z^k_{\mathrm{wm}}(v \mid c_t) = \bar{z}_T(v \mid c_t) + \lambda_{\mathrm{wm}} s^k_{\mathrm{wm}}(v \mid c_t), \tag{14}$$

where $\lambda_{\mathrm{wm}} \geq 0$ controls the overall watermark strength. The next-token distribution is obtained by softmax over C_t:

$$q_k(v \mid c_t) = \frac{\mathbb{I}[v \in C_t] \exp(z^k_{\mathrm{wm}}(v \mid c_t))}{\sum_{u \in C_t} \exp(z^k_{\mathrm{wm}}(u \mid c_t))}. \tag{15}$$

Top-p sampling is then applied to q_k. GGRW introduces a key-dependent soft bias that favors tokens beneficial for downstream watermark inheritance while preserving generation quality. The protected generation procedure is illustrated in Figure 2.

3.4 Held-Out Watermark Detection

Since protected sampling is guided by alignment scores, we evaluate watermark inheritance using a carrier statistic constructed from the same alignment criterion.

After a downstream student has been fine-tuned on protected teacher outputs, we evaluate watermark inheritance on a held-out set $\mathcal{E}_{\mathrm{det}} = \{(c_i, C_i)\}_{i=1}^N$, which is disjoint from the data used for watermark construction and student fine-tuning.

For each held-out context, we define the carrier weight as

$$w_{i,k}(v) = \mathbb{I}[v \in G^k_i][\Delta^k_{\mathrm{wm}}(v \mid c_i)]_+. \tag{16}$$

Let $p_m(v \mid c_i, C_i)$ denote the suspect model's next-token probability normalized over C_i. Its weighted carrier response is

$$W^m_{i,k} = \sum_{v \in C_i} p_m(v \mid c_i, C_i)\, w_{i,k}(v),$$

and $W^T_{i,k}$ is defined analogously for the teacher. We aggregate the teacher-centered excess response as

$$M^m_k = \frac{1}{N} \sum_{i=1}^N \left(W^m_{i,k} - W^T_{i,k}\right). \tag{17}$$

A positive value indicates that the suspect model assigns greater alignment-weighted probability mass than the teacher to target-key carriers on average.

To assess whether this response is specific to the target key, we sample B independent wrong keys $k'_1, \ldots, k'_B$. For each wrong key, we recompute the corresponding green partition and carrier weights while keeping the watermark direction, alignment scores, and model probabilities fixed, yielding the statistic $M^m_{k'_b}$. We compute the empirical one-sided significance as

$$p^m_{\mathrm{wm}} = \frac{1 + \sum_{b=1}^{B} \mathbb{I}[M^m_{k'_b} \geq M^m_k]}{B+1}. \tag{18}$$

We declare watermark inheritance when $M^m_k > 0$ and $p^m_{\mathrm{wm}} \leq \alpha$.

The positivity condition requires the suspect model to exhibit the intended teacher-relative shift, while the conditional wrong-key test assesses whether this response is specific to the target key under the fixed watermark direction.

4 Experiments

We conduct experiments to evaluate watermark inheritance, downstream utility, held-out detectability, and robustness to clean-data dilution. Specifically, we investigate whether GGRW can transfer watermark signals across different student architectures. We further evaluate whether protected fine-tuning preserves downstream model utility and whether the inherited watermark remains detectable on disjoint held-out contexts. Finally, we examine the robustness of GGRW under substantial clean-data dilution.

4.1 Experimental Setup

Models. We use DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025) as the teacher and Qwen2.5-3B (Yang et al., 2024) as the proxy throughout the main experiments. We consider three downstream students. Qwen2.5-3B provides a proxy-matched setting, while Llama-3.2-3B and Llama-3.2-3B-Instruct (Grattafiori et al., 2024) evaluate watermark inheritance across model families and tokenizers.

Data. We conduct experiments on the GSM8K dataset (Cobbe et al., 2021), which consists of grade school math problems. We use disjoint subsets of training set for watermark construction, protected teacher generation and downstream fine-tuning, and held-out watermark detection. Specifically, 256 examples are used to construct the watermark direction, 1024 examples are used for protected teacher generation and student fine-tuning, and 512 examples are reserved for held-out watermark detection. Teacher and student utility are measured by exact-answer accuracy on 512 examples from the GSM8K test set.

Training and generation. Each student is fine-tuned for two epochs using rank-8 LoRA adapters with a learning rate of 2×10^{-4}. We use temperature $\tau = 0.7$, top-$p = 0.95$, and finite-difference radius $\epsilon = 0.01$ for protected generation. To examine the effect of watermark strength on detectability and utility, we vary $\lambda_{\mathrm{wm}} \in \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$ while fixing $\mu_{\mathrm{wm}} = 0.5$. At $\lambda_{\mathrm{wm}} = 0$, we obtain an unwatermarked sampling control under the same decoding configuration.

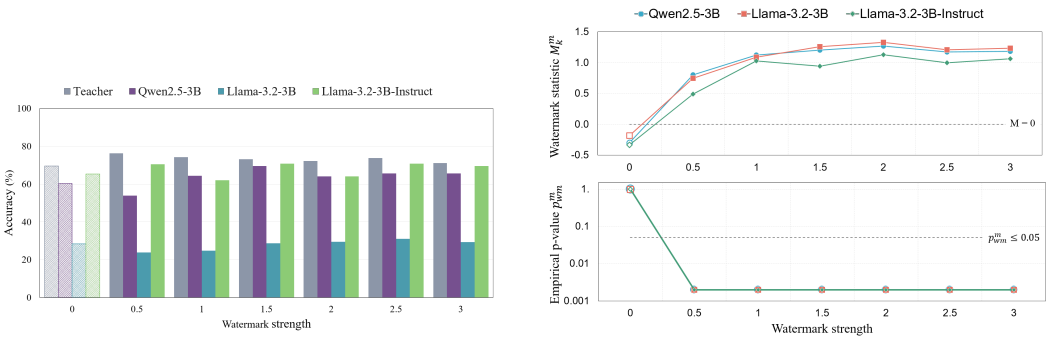
Evaluation. We evaluate watermark inheritance using the held-out detector introduced in Section 3.4. We report the teacher-centered statistic M^m_k and its empirical wrong-key significance p^m_{wm} using 500 wrong keys. We declare watermark inheritance when $M^m_k > 0$ and $p^m_{\mathrm{wm}} \leq 0.05$.

4.2 Watermark Effect Across Student Architectures

We present task utility and watermark inheritance on GSM8K across watermark strengths in Figure 3, and exact numerical results are provided in Appendix B.1. As illustrated in Figure 3a, task utility does not vary significantly with watermark strength. Across nonzero strengths, teacher accuracy remains between 71.09% and 76.37%, while student accuracy depends on both watermark strengths and the student backbone.

Figure 3b shows the corresponding held-out watermark detection results. For Qwen2.5-3B student, the $\lambda_{\mathrm{wm}} = 0$ control obtains a negative teacher-centered statistic ($M^m_k = -0.297$) and is not

detected. In contrast, every evaluated nonzero watermark strength obtains a positive response, with M_k^m ranging from 0.801 to 1.270 and $p_{\text{wm}}^m = 0.0020$ in all cases. The same conclusion persists for students whose architectures differ from the proxy. For Llama-3.2-3B student, the $\lambda_{\text{wm}} = 0$ control remains undetected ($M_k^m = -0.183$, $p_{\text{wm}}^m = 0.9880$), whereas all nonzero strengths are detected, with M_k^m ranging from 0.747 to 1.333. Llama-3.2-3B-Instruct exhibits the same pattern. The unwatermarked control is not detected, whereas every evaluated nonzero strength satisfies the detection criterion. These results indicate that the watermark constructed using the Qwen proxy is inherited not only by the proxy-matched student, but also by students from a different model family.



(a) Teacher and student utility.

(b) Held-out watermark detection.

Figure 3: **Task utility and watermark inheritance on GSM8K across watermark strengths.** (a) Teacher and student exact-answer accuracy. (b) Held-out detection using M_k^m and p_{wm}^m. Dashed lines indicate $M_k^m = 0$ and $p_{\text{wm}}^m = 0.05$; open and filled markers denote undetected and detected settings respectively.

4.3 Comparison with Existing Methods

We compare GGRW with KGW watermarking (Kirchenbauer et al., 2023) and Antidistillation Fingerprinting (ADFP) (Xu et al., 2026) under the setup in Section 4.1. Since the methods use different detection statistics and null hypotheses, we compare their prescribed detection decisions and utility. Implementation details and method-specific detection protocols are provided in Appendix B.2.

As shown in Table 1, none of the evaluated KGW configurations achieves detection under the prescribed protocol on disjoint held-out contexts. ADFP likewise remains undetected across all evaluated strengths under disjoint held-out contexts. Moreover, increasing the ADFP strength substantially reduces utility. At its strongest setting, teacher and student accuracy fall to 35.94% and 46.09%, respectively. In contrast, GGRW is detected at both representative settings reported in the table. At $\lambda_{\text{wm}} = 1.5$, GGRW achieves 73.24% teacher accuracy and 69.53% student accuracy, while $\lambda_{\text{wm}} = 2.5$ retains 73.83% and 65.62%, respectively. Thus, GGRW achieves held-out watermark detection under this evaluation setting, whereas the evaluated KGW and ADFP configurations do not. Appendix B.3 further analyzes the detectability of the ADFP scheme under the *same-prompt* setting.

We additionally compare GGRW with benchmark watermarking (BWM) (Sander et al., 2025), which inserts watermarks into rephrased benchmark questions and performs reading-mode detection directly on the watermarked benchmark inputs. GGRW instead embeds its signal in teacher-generated outputs and evaluates inheritance on held-out contexts. As shown in Table 2, we compare BWM and GGRW under the same unwatermarked held-out contexts. All BWM detection p-values exceed the 0.05 threshold under held-out evaluation. The corresponding GGRW students all satisfy the held-out detection criterion, with positive M_k^m and $p_{\text{wm}}^m = 0.0020$. Appendix B.4 further analyzes the detectability of the BWM scheme under the watermarked question setting.

ADFP becomes detectable under its same-prompt evaluation protocol, but remains undetected under the disjoint held-out evaluation considered above. Similarly, BWM students are detected when evaluated on the watermarked questions. These results highlight that our comparison focuses on whether inherited signals remain detectable under disjoint held-out evaluation. GGRW remains detectable under this held-out setting.

Table 1: **Comparison with existing fingerprinting methods on GSM8K.** Teacher and student accuracies are reported in percentages. The reported detection quantities are method-specific (KGW: z-score; ADFP: green-token probability, GTP; GGRW: M_k^m and p_{wm}^m). We report two representative detected settings for GGRW; its full strength sweep appears in Table B.1.

Method	Strength	Teacher acc.	Student acc.	Statistic	Det.
KGW	$\delta = 1.0$	72.85	66.41	$z = 1.169$	No
	$\delta = 1.5$	71.09	71.29	$z = 1.311$	No
	$\delta = 2.0$	71.88	67.38	$z = 1.347$	No
ADFP	$\lambda = 16$	72.46	58.98	GTP=0.510	No
	$\lambda = 64$	67.38	46.29	GTP=0.505	No
	$\lambda = 256$	35.94	46.09	GTP=0.513	No
GGRW (ours)	$\lambda_{\mathrm{wm}} = 1.5$	73.24	69.53	$M_k^m = 1.207,\ p_{\mathrm{wm}}^m = 0.0020$	**Yes**
	$\lambda_{\mathrm{wm}} = 2.5$	73.83	65.62	$M_k^m = 1.174,\ p_{\mathrm{wm}}^m = 0.0020$	**Yes**

Table 2: **Held-out comparison with benchmark watermarking (BWM).** BWM ($\delta = 4$) is evaluated on the original unwatermarked GSM8K held-out questions. The two methods use different detection statistics and null hypotheses.

	BWM ($\delta = 4$)		GGRW ($\lambda_{\mathrm{wm}} = 1.5$)		
Student	p	Det.	M_k^m	p_{wm}^m	Det.
Qwen2.5-3B	0.0509	No	1.207	0.0020	**Yes**
Llama-3.2-3B	0.2130	No	1.261	0.0020	**Yes**
Llama-3.2-3B-Instruct	0.1879	No	0.945	0.0020	**Yes**

4.4 Robustness to Clean-Data Dilution

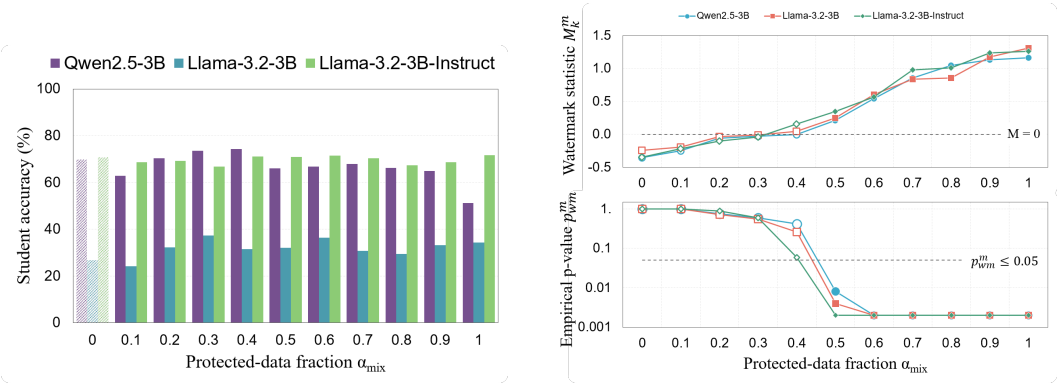
In practice, a downstream model may be fine-tuned on a mixture of protected and ordinary teacher-generated outputs. We therefore evaluate whether the inherited watermark remains detectable under clean-data dilution. We fix $\lambda_{\mathrm{wm}} = 1.5$ and keep the downstream training set fixed at 1024 examples. A fraction α_{mix} of the training examples is sampled from the protected pool, while the remaining $1 - \alpha_{\mathrm{mix}}$ is sampled from a disjoint pool of unprotected teacher outputs.

Figure 4 shows student utility across different protected-data fractions and the corresponding held-out watermark detection results. Exact numerical results are provided in Appendix B.5. As shown in Figure 4a, student accuracy does not exhibit a significant degradation as the protected-data fraction increases. Figure 4b shows that watermark detectability improves as the protected-data fraction increases. None of the three students satisfies the detection criterion when $\alpha_{\mathrm{mix}} \leq 0.4$, whereas all three satisfy the detection criterion when $\alpha_{\mathrm{mix}} \geq 0.5$. The teacher-centered watermark statistic M_k^m increases consistently with the protected-data fraction. At $\alpha_{\mathrm{mix}} = 0.6$, all three students obtain the minimum empirical p-value of 0.0020. These results indicate that the inherited watermark remains detectable under substantial clean-data dilution.

4.5 Ablation Study

We ablate the two components of the protected sampling objective at $\lambda_{\mathrm{wm}} = 1$: the gradient-weighted key-dependent reward term and the teacher-gap penalty term. Base Sampling disables both components, Gap Only retains only the teacher-gap penalty term, Key Only retains only the gradient-weighted key-dependent reward term, and Full uses both components.

As shown in Table 3, watermark inheritance mainly relies on the gradient-weighted key-dependent reward. Without this component, both Base Sampling and Gap Only yield a negative teacher-centered statistic ($M_k^m = -0.292$) with $p_{\mathrm{wm}}^m = 1.0000$, and neither variant satisfies the detection criterion. In contrast, Key Only produces a positive watermark statistic ($M_k^m = 1.096$) with $p_{\mathrm{wm}}^m = 0.0020$, while the full objective achieves a similar result ($M_k^m = 1.123$, $p_{\mathrm{wm}}^m = 0.0020$). Thus, removing the key-dependent reward prevents held-out watermark detection, whereas the key-dependent reward alone is sufficient to produce successful watermark detection.



(a) Student utility.

(b) Held-out watermark detection.

Figure 4: **Robustness to clean-data dilution on GSM8K.** (a) Student exact-answer accuracy across protected-data fractions α_{mix}. (b) Held-out watermark detection using M_k^m and p_{wm}^m. Dashed lines indicate $M_k^m = 0$ and $p_{\text{wm}}^m = 0.05$; open and filled markers denote undetected and detected settings, respectively.

Table 3: Ablation of the protected sampling objective at $\lambda_{\text{wm}} = 1$.

Variant	Components		Utility		Watermark detection		
	Gradient-weighted key-dependent reward	Teacher-gap penalty	Teacher acc.	Student acc.	M_k^m	p_{wm}^m	Det.
Base sampling	–	–	69.53	66.41	−0.292	1.0000	No
Gap only	–	✓	73.83	61.91	−0.292	1.0000	No
Key only	✓	–	73.83	67.19	1.096	0.0020	**Yes**
Full	✓	✓	74.22	64.45	1.123	0.0020	**Yes**

Comparing KEY ONLY with FULL, adding the gap penalty slightly increases teacher accuracy from 73.83% to 74.22% and M_k^m from 1.096 to 1.123, while student accuracy decreases from 67.19% to 64.45%. Moreover, GAP ONLY remains undetected despite improving teacher accuracy over BASE SAMPLING. These results indicate that the gradient-weighted key-dependent reward is the essential component for watermark inheritance. Meanwhile, the teacher-gap penalty has a limited effect on watermark detection and does not consistently improve downstream student utility.

5 CONCLUSION

In this paper, we introduced Gradient-Guided Radioactive Watermarking (GGRW), a generation-time framework that enables watermark inheritance after downstream fine-tuning. GGRW explicitly links token selection to downstream learning dynamics by favoring plausible teacher tokens whose induced updates align with a private watermark direction. Experiments on GSM8K demonstrate that the resulting watermark signal is inherited by both proxy-matched and cross-architecture students. The inherited watermark remains detectable even when protected traces are substantially diluted with clean teacher-generated data. Importantly, GGRW enables watermark detection on disjoint held-out contexts, while the evaluated baseline configurations do not satisfy the detection criterion under the same evaluation setting. These results suggest that explicitly optimizing watermarks for downstream inheritance provides a promising approach for provenance tracing of teacher-generated data.

6 AI USE STATEMENT

We used AI-assisted tools for language polishing, including grammar correction and improvements to clarity, readability, and academic expression. All AI-assisted edits were carefully reviewed and revised by the authors.

REFERENCES

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In Shipra Agrawal and Aaron Roth (eds.), *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 1125–1139. PMLR, 30 Jun–03 Jul 2024. URL https://proceedings.mlr.press/v247/christ24a.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, Ali Cemgil, Ahmed Zahra, Kitty Stacpoole, and Pushmeet Kohli. Scalable watermarking for identifying large language model outputs. *Nature*, 634:818–823, 10 2024. doi: 10.1038/s41586-024-08025-4.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. In *IEEE International Workshop on Information Forensics and Security, WIFS 2023, Nürnberg, Germany, December 4-7, 2023*, pp. 1–6. IEEE, 2023. doi: 10.1109/WIFS58808.2023.10374576. URL https://doi.org/10.1109/WIFS58808.2023.10374576.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *Int. J. Comput. Vis.*, 129(6):1789–1819, 2021. doi: 10.1007/S11263-021-01453-Z. URL https://doi.org/10.1007/s11263-021-01453-z.

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, 2024. URL https://arxiv.org/abs/2407.21783.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge Distillation of Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://mlanthology.org/iclr/2024/gu2024iclr-minillm/.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. 2015. URL http://arxiv.org/abs/1503.02531.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *International Conference on Learning Representations*, 2024. URL https://mlanthology.org/iclr/2024/hu2024iclr-unbiased/.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust Distortion-Free Watermarks for Language Models. *Transactions on Machine Learning Research*, 2024. URL https://mlanthology.org/tmlr/2024/kuditipudi2024tmlr-robust/.

Aiwei Liu, Leyi Pan, Xuming Hu, Shuang Li, Lijie Wen, Irwin King, and Philip Yu. An unforgeable publicly verifiable watermark for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 8291–8307, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/214d2cffc381938be6f7254d5382904f-Paper-Conference.pdf.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 6499–6519, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1a2131ebe25bd55e4fc734126ea583ed-Paper-Conference.pdf.

Yepeng Liu and Yuheng Bu. Adaptive text watermark for large language models. 235:30718–30737, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/liu24e.html.

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13228–13251, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.648. URL https://aclanthology.org/2025.acl-long.648/.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Herve Jegou. Radioactive data: tracing through training. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8326–8335. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/sablayrolles20a.html.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. 2024. URL https://arxiv.org/abs/2402.14904.

Tom Sander, Pierre Fernandez, Saeed Mahloujifar, Alain Durmus, and Chuan Guo. Detecting benchmark contamination through watermarking. 2025. URL https://arxiv.org/abs/2502.17259.

Yash Savani, Asher Trockman, Zhili Feng, Yixuan Even Xu, Avi Schwarzschild, Alexander Robey, Marc Finzi, and J. Zico Kolter. Antidistillation sampling. 2025. URL https://arxiv.org/abs/2504.13146.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. A resilient and accessible distribution-preserving watermark for large language models. 235:53443–53470, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wu24h.html.

Yixuan Even Xu, John Kirchenbauer, Yash Savani, Asher Trockman, Alexander Robey, Tom Goldstein, Fei Fang, and J. Zico Kolter. Antidistillation fingerprinting. 2026. URL https://arxiv.org/abs/2602.03812.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang,

Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Xin Yi, Yue Li, Shunfan Zheng, Linlin Wang, Xiaoling Wang, and Liang He. Unified attacks to large language model watermarks: Spoofing and scrubbing in unauthorized knowledge distillation. *Knowledge-Based Systems*, 329:114295, 2025. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2025.114295. URL https://www.sciencedirect.com/science/article/pii/S095070512501336X.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable Robust Watermarking for AI-Generated Text. In *ICML 2023 Workshops: DeployableGenerativeAI*, 2023. URL https://mlanthology.org/icmlw/2023/zhao2023icmlw-provable/.

A Additional Derivations

A.1 From Token-Level Fine-Tuning to the Watermark Directional Derivative

We provide the derivation of Eq. (6). For a candidate token v under context c, define its token-induced update direction as

$$d_v = \nabla_{\theta_P} \log p(v \mid c; \theta_P). \tag{19}$$

A single gradient-descent step on the negative log-likelihood of v therefore yields

$$\theta'_P(v) = \theta_P + \eta d_v, \tag{20}$$

where $\eta > 0$ is the fine-tuning step size.

Assuming that R_k is differentiable in a neighborhood of θ_P, its first-order Taylor expansion along d_v is

$$R_k(\theta_P + \eta d_v) = R_k(\theta_P) + \eta \langle \nabla_{\theta_P} R_k(\theta_P), d_v \rangle + O(\eta^2). \tag{21}$$

Subtracting $R_k(\theta_P)$ from both sides and dividing by η gives

$$\frac{R_k(\theta_P + \eta d_v) - R_k(\theta_P)}{\eta} = \langle \nabla_{\theta_P} R_k(\theta_P), d_v \rangle + O(\eta). \tag{22}$$

Taking $\eta \to 0$ eliminates the higher-order term:

$$\begin{aligned} \lim_{\eta \to 0} \frac{R_k(\theta_P + \eta d_v) - R_k(\theta_P)}{\eta} &= \langle \nabla_{\theta_P} R_k(\theta_P), d_v \rangle \\ &= \langle \nabla_{\theta_P} R_k(\theta_P), \nabla_{\theta_P} \log p(v \mid c; \theta_P) \rangle . \end{aligned} \tag{23}$$

Thus, the token-level quantity used by GGRW is precisely the directional derivative of the model-level watermark response along the parameter update that would be induced by fine-tuning on token v. Its sign determines whether that update locally increases or decreases the target-key response, while its magnitude measures the corresponding first-order effect.

A.2 Accuracy of the Symmetric Finite-Difference Approximation

We next justify the centered finite-difference approximation in Eq. (11). For a fixed token v and context c, define the scalar function

$$f_v(\epsilon) = \log p \left(v \mid c; \theta_P + \epsilon g^k_{\mathrm{wm}} \right). \tag{24}$$

Its derivative at $\epsilon = 0$ is

$$\begin{aligned} f'_v(0) &= \frac{\mathrm{d}}{\mathrm{d}\epsilon} \log p \left(v \mid c; \theta_P + \epsilon g^k_{\mathrm{wm}} \right) \Big|_{\epsilon=0} \\ &= \left\langle g^k_{\mathrm{wm}}, \nabla_{\theta_P} \log p(v \mid c; \theta_P) \right\rangle . \end{aligned} \tag{25}$$

Assuming sufficient smoothness, the Taylor expansions around $\epsilon = 0$ are

$$f_v(\epsilon) = f_v(0) + \epsilon f'_v(0) + \frac{\epsilon^2}{2} f''_v(0) + \frac{\epsilon^3}{6} f'''_v(0) + O(\epsilon^4), \tag{26}$$

$$f_v(-\epsilon) = f_v(0) - \epsilon f'_v(0) + \frac{\epsilon^2}{2} f''_v(0) - \frac{\epsilon^3}{6} f'''_v(0) + O(\epsilon^4). \tag{27}$$

Subtracting Eq. (27) from Eq. (26) gives

$$f_v(\epsilon) - f_v(-\epsilon) = 2\epsilon f'_v(0) + \frac{\epsilon^3}{3} f'''_v(0) + O(\epsilon^5). \tag{28}$$

Dividing by 2ϵ yields

$$\frac{f_v(\epsilon) - f_v(-\epsilon)}{2\epsilon} = f'_v(0) + \frac{\epsilon^2}{6} f'''_v(0) + O(\epsilon^4). \tag{29}$$

Therefore,

$$\Delta^k_{\mathrm{wm}}(v \mid c) = \left\langle g^k_{\mathrm{wm}}, \nabla_{\theta_P} \log p(v \mid c; \theta_P) \right\rangle + O(\epsilon^2). \tag{30}$$

Hence, the symmetric two-sided estimator used by GGRW has second-order truncation error in the perturbation radius ϵ. Importantly, the two evaluations at $\theta_P + \epsilon g^k_{\mathrm{wm}}$ and $\theta_P - \epsilon g^k_{\mathrm{wm}}$ cancel the leading even-order terms, making the approximation more accurate than a one-sided finite difference for sufficiently small ϵ.

B Additional Experimental Results

B.1 Full Results for the Watermark-Strength Sweep

Table B.1 reports the exact numerical results underlying Figure 3. We report teacher and student exact-answer accuracy, the teacher-centered watermark statistic M_k^m, the empirical wrong-key p-value p_{wm}^m, and the resulting detection decision for each watermark strength. A watermark is detected when $M_k^m > 0$ and $p_{\mathrm{wm}}^m \leq 0.05$. With 500 wrong keys, 0.0020 is the minimum attainable corrected empirical p-value.

Table B.1: Exact numerical results for the watermark-strength sweep shown in Figure 3. Accuracies are reported in percentages.

Proxy → student	λ_{wm}	Teacher acc.	Student acc.	M_k^m	p_{wm}^m	Detect
Qwen → Qwen	0	69.53	60.35	-0.297	1.0000	No
	0.5	76.37	53.91	0.801	0.0020	Yes
	1	74.22	64.45	1.123	0.0020	Yes
	1.5	73.24	69.53	1.207	0.0020	Yes
	2	72.27	64.06	1.270	0.0020	Yes
	2.5	73.83	65.62	1.174	0.0020	Yes
	3	71.09	65.62	1.184	0.0020	Yes
Qwen → Llama	0	69.53	28.32	-0.183	0.9880	No
	0.5	76.37	23.83	0.747	0.0020	Yes
	1	74.22	24.80	1.092	0.0020	Yes
	1.5	73.24	28.71	1.261	0.0020	Yes
	2	72.27	29.49	1.333	0.0020	Yes
	2.5	73.83	31.05	1.210	0.0020	Yes
	3	71.09	29.30	1.234	0.0020	Yes
Qwen → Llama-Instruct	0	69.53	65.43	-0.336	1.0000	No
	0.5	76.37	70.51	0.492	0.0020	Yes
	1	74.22	62.11	1.029	0.0020	Yes
	1.5	73.24	70.90	0.945	0.0020	Yes
	2	72.27	64.06	1.132	0.0020	Yes
	2.5	73.83	70.90	1.001	0.0020	Yes
	3	71.09	69.53	1.062	0.0020	Yes

B.2 Baseline Implementations and Detection Protocols

We summarize the baseline configurations used in Section 4.3. Since the compared methods use different watermarking objectives and detection protocols, their strength parameters and detection statistics are method-specific and are not directly comparable.

KGW. We implement the red–green-list watermark of Kirchenbauer et al. (2023) with green-list fraction $\gamma = 0.5$. The watermark adds a positive logit bias δ to green-list tokens during generation; we evaluate $\delta \in \{1.0, 1.5, 2.0\}$. Detection is performed on the disjoint held-out contexts using the corresponding green-token test after removing repeated bigram contexts. The reported z-score and detection decision follow the KGW statistical test.

ADFP. We implement Antidistillation Fingerprinting (Xu et al., 2026) with a bigram hash and green-list fraction $\gamma = 0.5$. ADFP perturbs the teacher logits as $\tilde{z} = z + \lambda\Delta_{\mathrm{ADS}}$, where λ controls the perturbation strength. We evaluate $\lambda \in \{16, 64, 256\}$. The student is trained on the same 1024 GSM8K prompts. The detection is evaluated on 512 disjoint held-out prompts using the open-weight green-token-probability (GTP) statistic and its one-sided Hoeffding test. This setting differs from the full ADFP training protocol. The additional results under ADFP's same-prompt evaluation are reported in Appendix B.3.

BWM. We adapt the benchmark-watermarking procedure of Sander et al. (2025) to GSM8K. In BWM, the teacher rephrases benchmark questions using a KGW-style watermark with context width

Table B.2: **ADFP detection under disjoint held-out and same-prompt evaluations.** Detection uses the one-sided Hoeffding test at $\alpha = 0.05$. The held-out protocol uses 512 disjoint prompt-level contexts. The same-prompt protocol evaluates independently sampled responses to the training prompts which contains substantially more token-level contexts.

		Disjoint held-out			Same-prompt		
Student	λ	GTP	p	Det.	GTP	p	Det.
Qwen2.5-3B	16	0.510	0.8985	No	0.504	0.1080	No
	64	0.505	0.9713	No	0.507	0.00105	**Yes**
	256	0.513	0.8320	No	0.509	1.72×10^{-6}	**Yes**
Llama-3.2-3B	16	0.502	0.9949	No	0.505	0.0530	No
	64	0.506	0.9682	No	0.506	0.0135	**Yes**
	256	0.501	0.9997	No	0.508	9.42×10^{-6}	**Yes**
Llama-3.2-3B-Instruct	16	0.495	1.0000	No	0.505	0.0301	**Yes**
	64	0.505	0.9766	No	0.507	6.41×10^{-4}	**Yes**
	256	0.505	0.9718	No	0.509	1.25×10^{-6}	**Yes**

$w = 2$, green-list fraction $\gamma = 0.5$, and watermark strength $\delta = 4$. For controlled comparison, we use a corresponding corpus generated with $\delta = 0$ as the unwatermarked control. Students are fine-tuned on 1024 rephrased question–answer examples in our main experiments. The main-text comparison evaluates BWM in reading mode on the 512 unwatermarked held-out GSM8K questions. We additionally evaluate its native watermarked-question settings in Appendix B.4.

B.3 Supplementary ADFP Evaluation

We further evaluate ADFP under the same-prompt protocols. ADFP students are trained on the same 1024 GSM8K teacher traces using rank-8 LoRA for two epochs. We use a bigram hash with green-list fraction $\gamma = 0.5$ and evaluate $\lambda \in \{16, 64, 256\}$.

The *disjoint held-out* protocol uses 512 GSM8K prompts that are used neither for fingerprinted generation nor for student fine-tuning. In contrast, the *same-prompt* protocol follows ADFP's open-weight evaluation. It reuses the 1024 training prompts but independently samples new fingerprinted responses using a different generation seed. Student green-token probability (GTP) is evaluated over the response-token contexts after globally removing repeated bigrams. Thus, the same-prompt evaluation does not reuse the original generated responses, but its prompts are not disjoint from those used for student fine-tuning.

Table B.2 shows a clear dependence on the evaluation protocol. Under the disjoint held-out protocol, none of the nine protected student–strength configurations is detected. In contrast, under the same-prompt evaluation, Qwen2.5-3B and Llama-3.2-3B are detected at $\lambda = 64$ and 256, while Llama-3.2-3B-Instruct is detected at all three evaluated strengths. These results show that the ADFP signal is detectable across student architectures under the same-prompt evaluation, whereas the corresponding configurations remain undetected on the disjoint held-out prompts used in our main comparison.

B.4 Supplementary BWM Evaluation

We additionally examine benchmark watermarking (BWM) under its watermarked-question protocol. Following the benchmark-rephrasing setup, DeepSeek-R1-Distill-Qwen-7B rephrases GSM8K questions using a KGW-style watermark with context width $w = 2$, green-list fraction $\gamma = 0.5$, and watermark strength $\delta = 4$. A corresponding $\delta = 0$ rephrasing corpus serves as the unwatermarked training control. Each student is trained on 1024 question–answer examples for two epochs using rank-8 LoRA.

BWM places the watermark in rephrased questions and applies the training loss to the full question–answer sequence. In contrast, GGRW embeds the watermark in teacher-generated answers, and the downstream student is fine-tuned on the corresponding protected outputs. Detection follows BWM's reading-mode detection procedure, which uses the actual watermarked question prompts as

Table B.3: **BWM reading-mode detection on watermarked GSM8K questions.** Base denotes the pretrained backbone, while $\delta = 0$ and $\delta = 4$ denote students trained on unwatermarked and watermarked rephrasing corpora, respectively. N is the number of eligible watermark contexts, and G/N denotes the fraction of green predictions. The reported p-values are one-sided binomial-tail probabilities.

		Watermarked training questions			Watermarked held-out questions		
Student	Model	N	G/N	p	N	G/N	p
Qwen2.5-3B	Base	29,112	0.4984	0.7148	16,487	0.5023	0.2819
	$\delta = 0$	29,113	0.5003	0.4622	16,488	0.5027	0.2488
	$\delta = 4$	29,115	0.5619	$\mathbf{1.39 \times 10^{-99}}$	16,486	0.5063	0.0534
Llama-3.2-3B	Base	28,803	0.5002	0.4761	16,301	0.5023	0.2808
	$\delta = 0$	28,622	0.4978	0.7697	16,166	0.5006	0.4465
	$\delta = 4$	28,594	0.5358	$\mathbf{5.29 \times 10^{-34}}$	16,160	0.5066	0.0468
Llama-3.2-3B-Instruct	Base	28,829	0.5008	0.3883	16,324	0.4992	0.5834
	$\delta = 0$	28,802	0.5015	0.3078	16,280	0.5014	0.3677
	$\delta = 4$	28,820	0.5428	$\mathbf{3.29 \times 10^{-48}}$	16,309	0.5021	0.2969

Table B.4: Exact numerical results for the clean-data dilution study shown in Figure 4. Accuracies are reported in percentages.

	Qwen2.5-3B			Llama-3.2-3B			Llama-3.2-3B-Instruct		
α_{mix}	M_k^m/p_{wm}^m	Acc.	Det.	M_k^m/p_{wm}^m	Acc.	Det.	M_k^m/p_{wm}^m	Acc.	Det.
0.0	$-0.354/1.0000$	69.73	No	$-0.246/1.0000$	26.76	No	$-0.351/1.0000$	70.70	No
0.1	$-0.248/1.0000$	62.89	No	$-0.193/0.9900$	24.22	No	$-0.223/0.9980$	68.75	No
0.2	$-0.062/0.7490$	70.31	No	$-0.038/0.7130$	32.23	No	$-0.105/0.8920$	69.14	No
0.3	$-0.027/0.5930$	73.63	No	$-0.011/0.5570$	37.30	No	$-0.040/0.5930$	66.80	No
0.4	$-0.008/0.4130$	74.22	No	$0.046/0.2610$	31.45	No	$0.152/0.0580$	71.09	No
0.5	$0.211/0.0080$	66.02	Yes	$0.247/0.0040$	32.03	Yes	$0.347/0.0020$	70.90	Yes
0.6	$0.542/0.0020$	66.80	Yes	$0.605/0.0020$	36.33	Yes	$0.569/0.0020$	71.48	Yes
0.7	$0.854/0.0020$	67.97	Yes	$0.835/0.0020$	30.66	Yes	$0.976/0.0020$	70.31	Yes
0.8	$1.044/0.0020$	66.21	Yes	$0.857/0.0020$	29.49	Yes	$1.006/0.0020$	67.38	Yes
0.9	$1.129/0.0020$	64.84	Yes	$1.177/0.0020$	33.20	Yes	$1.233/0.0020$	68.75	Yes
1.0	$1.164/0.0020$	51.17	Yes	$1.310/0.0020$	34.38	Yes	$1.261/0.0020$	71.68	Yes

input. For BWM, we follow its statistical testing protocol and use a significance level of $\alpha = 0.05$. Watermark detection is declared when the one-sided binomial-tail p-value satisfies $p < 0.05$.

As shown in Table B.3, all three students trained on the $\delta = 4$ rephrasing corpus exhibit strong detection on the watermarked training questions, while the corresponding base and $\delta = 0$ controls remain non-significant. This confirms that the adapted BWM implementation produces the intended contamination signal when evaluated on the watermarked training-question histories.

However, the signal transfers less consistently to unseen watermarked questions. On the 512 watermarked held-out rephrasings, the Qwen2.5-3B, Llama-3.2-3B, and Llama-3.2-3B-Instruct students yield $p = 0.0534$, 0.0468, and 0.2969, respectively. Thus, only Llama-3.2-3B crosses the 0.05 significance threshold, indicating limited transfer beyond the training-question histories.

Taken together, these results show that BWM produces strong contamination signals on watermark-bearing training questions but has limited transferability to unseen question contexts. The original-question evaluation used in our main comparison further extends BWM beyond its native watermark-bearing setting. Therefore, we use it to characterize held-out transfer under the specified protocol rather than to assess the effectiveness of BWM for its intended benchmark-contamination task.

B.5 Full Results for the Clean-Data Dilution Study

Table B.4 reports the exact numerical results underlying Figure 4. For each protected-data fraction α_{mix}, we report the teacher-centered watermark statistic M_k^m, the empirical wrong-key p-value p_{wm}^m, student exact-answer accuracy, and the resulting detection decision for all three student architectures. A watermark is detected when $M_k^m > 0$ and $p_{\text{wm}}^m \leq 0.05$. With 500 wrong keys, 0.0020 is the minimum attainable corrected empirical p-value.

C Examples of Watermarked Outputs

We present examples of protected teacher outputs generated by Gradient-Guided Radioactive Watermarking (GGRW), Antidistillation Fingerprinting (ADFP), and red–green-list watermarking (KGW). To facilitate comparison at comparable teacher utility, we select representative settings for which the teacher exact-answer accuracy on GSM8K exceeds 70%. Specifically, we show $\lambda_{\text{wm}} \in \{1.5, 3\}$ for GGRW, $\lambda = 16$ for ADFP, and $\delta = 1.0$ for KGW. The corresponding quantitative results are reported in Table 1 and Table B.1.

For each setting, we present two representative outputs, enabling a direct comparison of the resulting reasoning traces and answer quality.

Question 1

On a road map, 1/4 inch represents 8 miles of actual road distance. The towns of Pence and Hillcrest are represented by points 3 3/8 inches apart on the map. What is the actual distance, in miles, between the towns?

GGRW ($\lambda_{wm} = 1.5$)

First, I need to determine the actual distance represented by one inch on the map. Since 1/4 inch represents 8 miles, multiplying both by 4 gives 1 inch representing 32 miles.

Next, I'll convert the distance between Pence and Hillcrest on the map from 3 3/8 inches to an improper fraction, which is 27/8 inches.

Finally, I'll multiply the actual distance per inch (32 miles) by the map distance (27/8 inches) to find the real-world distance. This calculation gives 108 miles.
</think>

To determine the actual distance between the towns of Pence and Hillcrest, follow these steps:

Step 1: Understand the Scale
The map scale indicates that:
\[
\frac{1}{4} \text{ inch} = 8 \text{ miles}
\]
To find out how many miles **1 inch** represents, multiply both sides by 4:
\[
1 \text{ inch} = 8 \times 4 = 32 \text{ miles}
\]

Step 2: Convert the Map Distance to Inches
The towns are represented by points that are:
\[
3 \frac{3}{8} \text{ inches apart}
\]
Convert the mixed number to an improper fraction:
\[
3 \frac{3}{8} = \frac{27}{8} \text{ inches}
\]

Step 3: Calculate the Actual Distance
Multiply the number of inches by the miles per inch to find the actual distance:
\[
\frac{27}{8} \text{ inches} \times 32 \text{ miles per inch} = \frac{27 \times 32}{8} \text{ miles}
\]
Simplify the calculation:
\[
\frac{27 \times 32}{8} = 27 \times 4 = 108 \text{ miles}
\]

Final Answer
The actual distance between the towns of Pence and Hillcrest is:
\[
\boxed{108 \text{ miles}}
\]

GGRW ($\lambda_{wm} = 3$)

First, I need to determine the actual distance represented by one inch on the map. Since 1/4 inch represents 8 miles, multiplying both by 4 gives 1 inch representing 32 miles.

Next, I'll convert the distance between Pence and Hillcrest on the map from 3 3/8 inches to an improper fraction, which is 27/8 inches.

Finally, I'll multiply the actual distance per inch (32 miles) by the map distance (27/8 inches) to find the actual distance between the towns.
</think>

To determine the actual distance between the towns of Pence and Hillcrest, follow these steps:

1. **Understand the Scale:**

   - **Map Distance:** \(\frac{1}{4} \) inch represents **8 miles** of actual distance.

2. **Calculate Miles per Inch:**

   - If \(\frac{1}{4} \) inch = 8 miles,
   - Then 1 inch = \(8 \times 4 = 32 \) miles.

3. **Convert Map Distance to Improper Fraction:**

   - The towns are \(3 \frac{3}{8} \) inches apart on the map.
   - Convert to an improper fraction: \(3 \frac{3}{8} = \frac{27}{8} \) inches.

4. **Calculate Actual Distance:**

   - Actual distance = Map distance \(\times \) Miles per inch
   - Actual distance = \(\frac{27}{8} \) inches \(\times 32 \) miles/inch
   - Actual distance = \(\frac{27 \times 32}{8} \) miles
   - Actual distance = \(\frac{864}{8} \) miles
   - Actual distance = **108 miles**

\(\boxed{108}\) miles

Question 1

On a road map, 1/4 inch represents 8 miles of actual road distance. The towns of Pence and Hillcrest are represented by points 3 3/8 inches apart on the map. What is the actual distance, in miles, between the towns?

ADFP ($\lambda = 16$)

To determine the actual distance between the towns of Pence and Hillcrest, I will start by understanding the scale of the map. The problem states that 1/4 inch on the map corresponds to 8 miles in real life. This means that each inch on the map represents 8 miles multiplied by 4, which equals 32 miles.

Next, I need to calculate the distance between the towns on the map. The towns are 3 3/8 inches apart. To make calculations easier, I will convert this mixed number to an improper fraction. Multiplying the whole number by the denominator and adding the numerator, I get $(3 \times 8 + 3) = 27$. So, 3 3/8 inches is equivalent to 27/8 inches.

Now, to find the actual distance, I will multiply the map distance by the scale factor. This means multiplying 27/8 inches by 32 miles per inch. Performing the calculation, 27×32 equals 864, and then dividing by 8 gives 108.

Therefore, the actual distance between the towns of Pence and Hillcrest is 108 miles.
</think>

To determine the actual distance between the towns of **Pence** and **Hillcrest**, we'll follow a step-by-step approach based on the given scale of the map.

Step 1: Understand the Scale
The map uses the following scale:
- **1/4 inch** represents **8 miles**.
Step 2: Determine the Scale Factor
To find the distance represented by **1 inch** on the map:
\[
1\text{ inch} = 4 \times \frac{1}{4}\text{ inch} = 4 \times 8 \text{ miles} = 32\text{ miles}
\]

Step 3: Measure the Distance on the Map
The distance between the towns on the map is given as:
\[
3\frac{3}{8}\text{ inches}
\]

Convert the mixed number to an improper fraction:
\[
3\frac{3}{8} = \frac{27}{8}\text{ inches}
\]

Step 4: Calculate the Actual Distance
Multiply the map distance by the scale factor:
\[
\frac{27}{8}\text{ inches} \times 32\text{ miles per inch} = \frac{27 \times 32}{8}\text{ miles} = \frac{864}{8}\text{ miles} = 108\text{ miles}
\]

Final Answer
The actual distance between the towns of **Pence** and **Hillcrest** is:
\[
\boxed{108}
\]

KGW ($\delta = 1.0$)

First, I need to determine how many miles one inch on the map represents. Since 1/4 inch equals 8 miles, multiplying both sides by 4 shows that 1 inch equals 32 miles.

Next, I'll convert the distance between Pence and Hillcrest on the map, which is 3 3/8 inches, into an improper fraction for easier calculation. 3 3/8 inches is equivalent to 27/8 inches.

Finally, I'll multiply the number of inches (27/8) by the miles per inch (32) to find the actual distance. This calculation gives me 108 miles.
</think>

To determine the actual distance between the towns of Pence and Hillcrest, we'll follow these steps:

1. **Find the Scale of the Map:**

   The map states that \(\frac{1}{4} \) inch represents 8 miles. To find out how many miles 1 inch represents, we can set up a proportion:

   \[
   \frac{1}{4} \text{ inch} = 8 \text{ miles}
   \]

   Multiplying both sides by 4 to find the miles for 1 inch:

   \[
   1 \text{ inch} = 8 \times 4 = 32 \text{ miles}
   \]

2. **Convert the Map Distance to Inches:**

   The towns are 3 \(\frac{3}{8} \) inches apart on the map. It's easier to work with improper fractions, so let's convert 3 \(\frac{3}{8} \) inches to an improper fraction:

   \[
   3 \frac{3}{8} = \frac{24}{8} + \frac{3}{8} = \frac{27}{8} \text{ inches}
   \]

3. **Calculate the Actual Distance:**

   Now, multiply the number of inches by the miles per inch to find the actual distance:

   \[
   \text{Actual Distance} = \frac{27}{8} \text{ inches} \times 32 \text{ miles/inch} = \frac{27 \times 32}{8} \text{ miles}
   \]

   Simplify the calculation:

   \[
   \frac{27 \times 32}{8} = 27 \times 4 = 108 \text{ miles}
   \]

Final Answer:

\[
\boxed{108 \text{ miles}}
\]

Question 1

Hannah sold 40 pieces of cookies for $0.8 each and 30 cupcakes for $2 each. She used the money to buy 2 sets of measuring spoons for $6.5 each. How much money does she have left?

GGRW ($\lambda_{wm} = 1.5$)

First, I need to calculate the total revenue from selling cookies and cupcakes.

Hannah sold 40 cookies at $0.80 each. Multiplying 40 by 0.80 gives $32.00 from cookies.

She also sold 30 cupcakes at $2.00 each. Multiplying 30 by 2.00 gives $60.00 from cupcakes.

Adding both amounts, the total revenue is $32.00 + $60.00 = $92.00.

Next, I'll calculate the total cost of the measuring spoons. Each set costs $6.50, and she bought 2 sets. Multiplying 2 by 6.50 gives $13.00.

Finally, subtracting the cost of the measuring spoons from the total revenue, $92.00 - $13.00, leaves Hannah with $79.00.
</think>

Certainly! Let's break down the problem step by step to find out how much money Hannah has left after her purchases.

Step 1: Calculate the Total Revenue from Cookies

Hannah sold:
- **Cookies:** 40 pieces at \$0.80 each

\[
\text{Revenue from cookies} = 40 \times 0.80 = \$32.00
\]

Step 2: Calculate the Total Revenue from Cupcakes

Hannah sold:
- **Cupcakes:** 30 pieces at \$2.00 each

\[
\text{Revenue from cupcakes} = 30 \times 2.00 = \$60.00
\]

Step 3: Calculate the Total Revenue

\[
\text{Total Revenue} = \text{Revenue from cookies} + \text{Revenue from cupcakes} = 32.00 + 60.00 = \$92.00
\]

Step 4: Calculate the Total Cost of Measuring Spoons

Hannah bought:
- **Measuring spoons:** 2 sets at \$6.50 each

\[
\text{Total Cost} = 2 \times 6.50 = \$13.00
\]

Step 5: Calculate the Money Left After Purchases

\[
\text{Money Left} = \text{Total Revenue} - \text{Total Cost} = 92.00 - 13.00 = \$79.00
\]

Final Answer

Hannah has \(\boxed{\$79.00}\) left.

GGRW ($\lambda_{wm} = 3$)

First, calculate the total revenue from selling cookies by multiplying the number of cookies (40) by the price per cookie ($0.80).

Next, calculate the total revenue from selling cupcakes by multiplying the number of cupcakes (30) by the price per cupcake ($2.00).

Add the revenues from cookies and cupcakes to find the total revenue.

Then, calculate the total cost of the measuring spoons by multiplying the number of sets (2) by the price per set ($6.50).

Finally, subtract the total cost of the measuring spoons from the total revenue to determine the remaining money.
</think>

Let's solve the problem step by step.

1. Calculate the total revenue from selling cookies:
\[
40 \text{ cookies} \times \$0.80 \text{ per cookie} = \$32.00
\]

2. Calculate the total revenue from selling cupcakes:
\[
30 \text{ cupcakes} \times \$2.00 \text{ per cupcake} = \$60.00
\]

3. Add both revenues to find the total money Hannah earned:
\[
\$32.00 + \$60.00 = \$92.00
\]

4. Calculate the total cost of the measuring spoons:
\[
2 \text{ sets} \times \$6.50 \text{ per set} = \$13.00
\]

5. Subtract the total cost of measuring spoons from the total revenue to find the remaining money:
\[
\$92.00 - \$13.00 = \$79.00
\]

Final Answer:
\[
\boxed{\$79.00}
\]

Question 1

Hannah sold 40 pieces of cookies for $0.8 each and 30 cupcakes for $2 each. She used the money to buy 2 sets of measuring spoons for $6.5 each. How much money does she have left?

ADFP ($\lambda = 16$)

First, calculate the total money Hannah earned from selling cookies by multiplying the number of cookies by the price per cookie.

Next, calculate the total money from cupcakes by multiplying the number of cupcakes by their price.

Add both totals to find the overall money earned.

Subtract the cost of the measuring spoons (multiplied by two sets) from the total money earned to determine the remaining amount.
</think>

Solution:

Let's solve the problem step by step.

1. **Calculate the money earned from selling cookies:**

\[
\text{Money from cookies} = \text{Number of cookies} \times \text{Price per cookie}
\]

\[
\text{Money from cookies} = 40 \times \$0.8 = \$32
\]

2. **Calculate the money earned from selling cupcakes:**

\[
\text{Money from cupcakes} = \text{Number of cupcakes} \times \text{Price per cupcake}
\]

\[
\text{Money from cupcakes} = 30 \times \$2 = \$60
\]

3. **Calculate the total money earned:**

\[
\text{Total money earned} = \text{Money from cookies} + \text{Money from cupcakes}
\]

\[
\text{Total money earned} = \$32 + \$60 = \$92
\]

4. **Calculate the total cost of the measuring spoons:**

\[
\text{Total cost of measuring spoons} = \text{Number of sets} \times \text{Cost per set}
\]

\[
\text{Total cost of measuring spoons} = 2 \times \$6.5 = \$13
\]

5. **Calculate the money left after purchasing the measuring spoons:**

\[
\text{Money left} = \text{Total money earned} - \text{Total cost of measuring spoons}
\]

\[
\text{Money left} = \$92 - \$13 = \$79
\]

Final Answer:

\[
\boxed{79}
\]

KGW ($\delta = 1.0$)

First, I need to calculate the total earnings from selling cookies and cupcakes.

Hannah sold 40 cookies at $0.80 each. Multiplying 40 by 0.80 gives $32.

Next, she sold 30 cupcakes at $2 each. Multiplying 30 by 2 gives $60.

Adding both amounts, Hannah earned a total of $92.

Then, I'll calculate the total cost of the measuring spoons. Each set costs $6.50, and she bought 2 sets. Multiplying 2 by 6.50 gives $13.

Finally, subtracting the cost of the measuring spoons from her total earnings, Hannah has $79 left.
</think>

Sure, let's solve the problem step by step.

Problem Statement:
Hannah sold 40 pieces of cookies for \$0.8 each and 30 cupcakes for \$2 each. She used the money to buy 2 sets of measuring spoons for \$6.5 each. How much money does she have left?

Solution:

1. **Calculate the total earnings from cookies:**
\[
40 \text{ cookies} \times \$0.8 \text{ per cookie} = 40 \times 0.8 = \$32
\]

2. **Calculate the total earnings from cupcakes:**
\[
30 \text{ cupcakes} \times \$2 \text{ per cupcake} = 30 \times 2 = \$60
\]

3. **Find the total amount Hannah earned:**
\[
\$32 \text{ (from cookies)} + \$60 \text{ (from cupcakes)} = \$92
\]

4. **Calculate the total cost of the measuring spoons:**
\[
2 \text{ sets} \times \$6.5 \text{ per set} = 2 \times 6.5 = \$13
\]

5. **Determine the remaining money after purchasing the measuring spoons:**
\[
\$92 \text{ (total earnings)} - \$13 \text{ (cost of measuring spoons)} = \$79
\]

Final Answer:
\[
\boxed{79}
\]