LANGUAGE MODEL FINGERPRINTING REQUIRES RETHINKING WATERMARK TEACHERS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM fingerprinting via watermark distillation embeds a statistical watermark signal into model weights, enabling model owners to identify their models behind black-box APIs. Revisiting a recent protocol, we find that its utility evaluation understates text quality degradation in open-ended generation, favoring overly strong watermark teachers. Weakening the watermark improves text quality but sacrifices detectability. To move beyond this trade-off, we rethink whether text watermarks designed for verifying generated text are suitable distillation teachers for model fingerprinting. Such watermarks are typically designed to remain detectable from an individual output, limiting how sparse the watermark signal can be. In contrast, fingerprint verification can aggregate signal across queries, making sparser watermark signals viable. This raises a key question: where should the sparse signal be placed? We analyze signal placement through token surprisal and show that, even at comparable watermark strength, different placements can target tokens with different plausibility under the base model. This motivates near-tie restriction, which uses top-1-relative logit gaps to restrict the watermark bias to tokens close to the base model's top prediction. Across multiple models, near-tie improves detection–quality frontiers under deployment changes, preserves higher text quality across query budgets, and further improves existing watermarking schemes when combined with them.

1 INTRODUCTION

The strong performance and widespread adoption of large language models (LLMs) have intensified the competition to develop increasingly capable models, with a growing number of such models released with open weights (Minaee et al., 2024; Qwen Team, 2025). Once released, however, these models may be deployed behind black-box APIs without honoring restrictive licenses, raising concerns about model ownership and unauthorized use (White et al., 2024; Xu et al., 2024a). To build a healthy ecosystem, there is a growing need for reliable model provenance verification.

Recently, black-box model fingerprinting methods (Xu et al., 2024a; Russinovich & Salem, 2024; Nasery et al., 2025b; Gloaguen et al., 2026) have been proposed for this purpose. They broadly fall into backdoor-based (Xu et al., 2024a; Russinovich & Salem, 2024; Nasery et al., 2025b) and watermark-based (Gloaguen et al., 2026) approaches. In this work, we focus on watermark-based fingerprinting, which uses a text watermark originally designed to verify the provenance of generated text (Kirchenbauer et al., 2023) as a distillation teacher. Its watermarked output distribution is distilled into model weights on a designated fingerprint domain, such as French, so that queries from this domain elicit a detectable statistical signal. Compared with backdoor-based approaches, this framework offers three advantages: (i) statistical detection with false-positive control, (ii) natural-looking outputs that are challenging to detect and filter out, and (iii) robustness to deployment transformations such as decoding variations, quantization, pruning, and fine-tuning.

Despite these advantages, we find that the utility evaluation of a recent protocol substantially understates degradation in open-ended generation within the fingerprint domain. This favors overly strong watermark teachers that degrade generation quality for legitimate users in that domain. Reducing watermark strength improves text quality but sacrifices detectability, while more advanced watermarking schemes provide only limited gains when used as distillation teachers (Figure 3).

To move beyond this trade-off, we ask whether text watermarks are well suited as distillation teachers for model fingerprinting. Text watermarks (Kirchenbauer et al., 2023; Lee et al., 2024; Wang

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.

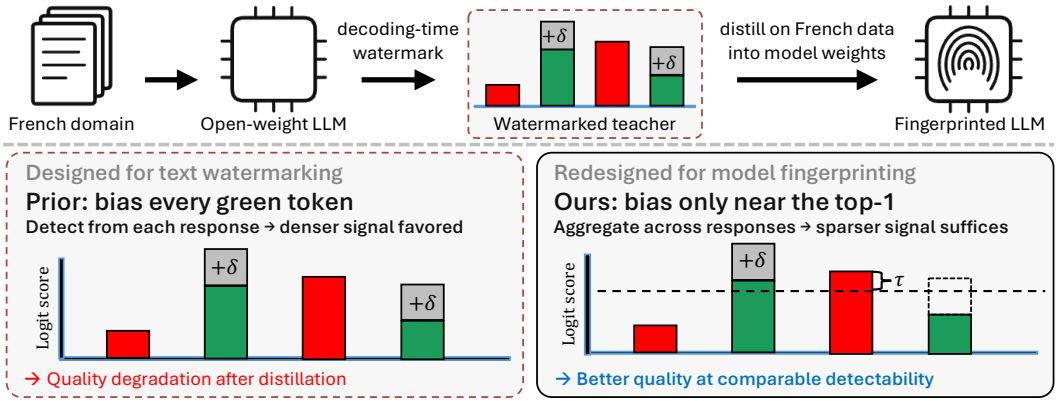


Figure 1: Overview of rethinking watermark teachers for model fingerprinting. Top: watermark-based fingerprinting distills a text watermark into model weights on a semantic domain such as French. The watermark splits tokens into green and red sets and biases green tokens to create a detectable statistical signal. Bottom: prior teachers bias every green token for individual-response detection, whereas fingerprint verification aggregates evidence across responses and permits sparser signals. Our near-tie restriction applies the bias only to green tokens near the top-1 logit.

et al., 2025b; Wu et al., 2023; Liu et al., 2024; Kuditipudi et al., 2023) are typically designed to remain detectable from an individual output, including short or modified outputs, which constrains how sparse their signal can be. Model fingerprinting, by contrast, can aggregate signal across multiple queries, making sparser teacher signals viable. This raises a key question: where should the remaining signal be placed? We analyze signal placement through token surprisal and show that, even at comparable watermark strength, different placements target tokens of different plausibility. Motivated by this analysis, we propose near-tie restriction, which uses logit gaps to limit teacher modifications to tokens close to the model's top prediction. Since it only restricts which tokens carry the watermark signal, near-tie can be incorporated into existing watermarking schemes without altering their underlying mechanisms.

Empirically, this simple restriction substantially improves detection–quality frontiers across multiple models and watermarking schemes under common deployment transformations. It also yields higher generation quality across query budgets, and further improves existing watermarking schemes when combined with them.

Our main contributions are summarized as follows:

- We show that the evaluation of a recent watermark-based fingerprinting protocol understates text quality degradation in open-ended generation and favors overly strong teachers.
- We point out an overlooked implication of model fingerprinting: aggregating signal across queries makes sparser teacher signals viable. Motivated by token-surprisal analysis, we propose near-tie restriction using base-model logit gaps for signal placement.
- Our comprehensive experiments show that near-tie restriction consistently improves detection–quality frontiers, preserves higher generation quality across query budgets, and further improves existing watermarking schemes when combined with them.

2 Related Work

LLM fingerprinting. White-box fingerprinting methods (Refael et al., 2024; Zeng et al., 2024; Zhang et al., 2025) provide robust verification but require access to model weights, limiting their practicality in real-world deployments. This motivates black-box methods, which verify ownership from queries and responses alone and broadly follow either a backdoor-based or watermark-based approach. Backdoor-based methods fine-tune a model to emit predetermined responses to secret trigger queries and verify ownership through exact matches (Xu et al., 2024a; Nasery et al., 2025b; Russinovich & Salem, 2024). However, atypical query–response pairs can undermine stealthiness, while exact-match verification can be fragile to deployment transformations. Watermark-based fingerprinting instead distills a watermark into model weights, enabling statistical verification from queries in a semantic domain and improving robustness to deployment transformations (Gloaguen

et al., 2026). However, we find that its evaluation understates text quality degradation on open-ended generation within the fingerprint domain. We provide extended related work in Appendix A.

LLM watermarking. LLM watermarking supports text provenance by embedding a hidden statistical signal into generated text. KGW (Kirchenbauer et al., 2023), a widely used red–green scheme, partitions the vocabulary and biases green-token logits during decoding, making green tokens statistically overrepresented. Subsequent work develops alternative partitioning and biasing strategies to improve the detection–quality trade-off (Lee et al., 2024; Wang et al., 2025b; Wu et al., 2023; Liu et al., 2024). Beyond decoding-time watermarking, Gu et al. (2024) show that watermarks can be distilled into open-weight models, enabling watermarked generation without modifying inference-time decoding. Building on this idea, Gloaguen et al. (2026) distill a semantically conditioned watermark for model fingerprinting. However, this directly inherits teachers designed for text provenance detection, overlooking that fingerprint verification aggregates evidence across responses. We exploit this distinction to design a sparser teacher that better preserves text quality.

3 PRELIMINARY

A language model $\mathcal{M}$ generates text by predicting the next token x_t in a sequence. Given the previous tokens $x_{<t}$, it outputs a vector of logits $\boldsymbol{\ell}_t$ over the vocabulary $\mathcal{V}$, which defines the next-token distribution $Q_t = \mathrm{softmax}(\boldsymbol{\ell}_t)$. The next token x_t is then randomly sampled according to Q_t depending on the choice of the decoding strategy.

Watermarking algorithm. LLM watermarking enables model owners to statistically verify whether a text was generated by their model or not. A common technique is to embed a detectable signal during decoding. In this work, we focus on KGW introduced by Kirchenbauer et al. (2023), which is a widely used red–green watermarking scheme. At each decoding step, a pseudorandom function of the secret key ξ and the preceding h tokens partitions the vocabulary $\mathcal{V}$ into a green list G_t, with $|G_t| = \gamma|\mathcal{V}|$, and its red complement. KGW adds a bias $\delta \geq 0$ to the logits of green tokens, yielding

$$\widetilde{\ell}_t(v) = \ell_t(v) + \delta \mathbf{1}\{v \in G_t\}, \text{ for all } v \in \mathcal{V}, \text{ and} \qquad \widetilde{Q}_t = \mathrm{softmax}(\widetilde{\boldsymbol{\ell}}_t)\ ,$$

where we omit the dependence of G_t on $[x_{t-h-1}, \ldots, x_{t-1}]$ and ξ for brevity. For a generated response $x = [x_0, \ldots, x_{L-1}]$, let $N_G(x) = \sum_{t=0}^{L-1} \mathbf{1}\{x_t \in G_t\}$ denote the number of generated green tokens. Under the null hypothesis of unwatermarked generation, the expected green-token count is γL. The detector therefore computes a test statistic,

$$z(x) \;=\; \frac{N_G(x) - \gamma L}{\sqrt{\gamma(1-\gamma)L}}\ , \tag{1}$$

and classifies the response as watermarked if $z(x) > \rho$, where ρ denotes the detection threshold.

Watermark-based fingerprinting. Watermark-based fingerprinting distills a decoding-time watermark into model weights. The distilled model emits the watermark signal without the need for modified decoding at inference time, thereby enabling black-box ownership verification (Gloaguen et al., 2026). Unlike LLM watermarking for text provenance, which is required to watermark every output text, model fingerprinting only needs to watermark when prompted by the verifier. This is achieved by conditioning the fingerprint on a secret semantic domain $\mathcal{D}_{\mathrm{fp}}$, e.g., French, and a fingerprinted model $\mathcal{M}_{\mathrm{fp}}(\widetilde{Q})$ is trained to match a watermarked teacher $\widetilde{Q}$ within this domain, minimizing

$$\mathcal{L}_{\mathrm{fp}}(\theta) \;=\; \mathbb{E}_{x \sim \mathcal{D}_{\mathrm{fp}}} \left[\sum_{t=1}^{|x|} \mathrm{KL}\left(\widetilde{Q}_t \| P_{\theta,t} \right) \right], \tag{2}$$

where $P_{\theta,t} = P_\theta(\cdot \mid x_{<t})$ denotes the distilled next-token distribution of $\mathcal{M}_{\mathrm{fp}}(\widetilde{Q})$ parametrized by θ and $KL(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence. A regularization term is additionally used to suppress watermarked generations outside $\mathcal{D}_{\mathrm{fp}}$ to limit the degradation of the model.

For verification, the owner issues N queries from $\mathcal{D}_{\mathrm{fp}}$ and concatenates their responses as $y^{1:N} = y^1 \circ \cdots \circ y^N$. Since similar queries can produce overlapping responses, aggregating all tokens may count the same watermark evidence multiple times and artificially inflate the detection statistic. Therefore, duplicate h-grams are globally removed by retaining only their first occurrences, yielding the scoring sequence $w = \mathrm{Dedup}_h(y^{1:N})$. However, each retained token is evaluated in its original

Table 1: Utility re-evaluation of the Llama-3.1-8B fingerprinting setting from Gloaguen et al. (2026), which selects KGW with $\delta = 4$ as the teacher distribution. French Benchmark (FB) accuracy and mean token entropy understate generation-quality degradation, whereas PPL and LLM-judge scores on WritingPrompts reveal substantial degradation in open-ended generation.

Model	FB Acc ↑	Mean Entropy[1]	PPL ↓	LLM Judge ↑
Base	0.64	2.15	4.63	7.76
Fingerprinted	0.61	3.08	**28.27**	**3.73**

response context when checking whether it is green. The detector then computes $z(w)$ using Eq. (1). The full training objective is provided in Appendix B.

Threat model. We consider a black-box ownership-verification setting in which the owner retains the base model $\mathcal{M}_{\text{base}}$ and a secret key ξ but can access a suspect model only by issuing queries and observing its responses, which may have gone through an unknown transformation $T \in \mathcal{T}$, where $\mathcal{T}$ includes the identity transformation, different sampling parameters, quantization, pruning, and fine-tuning. Let $Y_T^{1:N}$ denote the sampled responses obtained by issuing N fingerprint queries to $T(\mathcal{M}_{\text{fp}}(\widetilde{Q}))$. We define the worst-case expected z-score as

$$z_N^{\text{worst}}\left(\mathcal{M}_{\text{fp}}(\widetilde{Q})\right) \;=\; \min_{T\in\mathcal{T}} \mathbb{E}\big[z\left(Y_T^{1:N}\right)\big]\,. \tag{3}$$

We say a fingerprinted model $\mathcal{M}_{\text{fp}}(\widetilde{Q})$ is robustly detectable if the empirical estimate of z_N^{worst} exceeds a threshold $\rho > 0$.

4 Revisiting Utility Evaluation of LLM Fingerprinting

Gloaguen et al. (2026) instantiate their fingerprinting framework with French as the fingerprint domain $\mathcal{D}_{\text{fp}}$. Although the watermark behavior is intended to remain within $\mathcal{D}_{\text{fp}}$, preserving generation quality in this domain remains important for legitimate users, for example, using French. We revisit their utility evaluation in open-ended generation and examine their detectability-quality trade-off.

Prior evaluation understates generation-quality degradation. For KGW, Gloaguen et al. (2026) fix $\gamma = 0.25$, $h = 1$, and select $\delta = 4$ from a sweep over bias strengths, reporting high detectability with little utility degradation across the evaluated models. Their utility evaluation reports accuracy on the French Benchmark[2] and a metric labeled as PPL. However, this accuracy-based evaluation, such as multiple-choice tasks, does not capture quality degradation in open-ended generation. Moreover, we find that the reported PPL is computed as mean token entropy rather than the standard perplexity. We reproduce their Llama-3.1-8B-Instruct setting and evaluate it on 500 French-translated WritingPrompts examples (Fan et al., 2018). This open-ended setting admits many plausible next-token alternatives, making it more sensitive to watermark-induced probability shifts. As shown in Table 1, French Benchmark accuracy decreases only from 0.64 to 0.61, while PPL increases from 4.63 to 28.27 and the LLM-judge score (Appendix J) decreases from 7.76 to 3.73, revealing substantial generation-quality degradation.

A weaker bias partially reduces degradation at the cost of detectability. A direct way to reduce text quality degradation is to weaken the teacher's logit bias, thereby limiting the distortion of the teacher distribution. We therefore sweep $\delta \in \{1, 2, 4\}$ and evaluate each setting using the PPL ratio $\text{PPL}_{\text{fp}}/\text{PPL}_{\text{base}}$ for text quality and the worst-case z-score z_N^{worst} over $\mathcal{T}$ for robust detectability. The experimental setup, including transformations $\mathcal{T}$, is detailed in Section 6.1. As shown in Figure 2, reducing δ from 4 to 2 lowers the PPL ratio while maintaining robust detectability. However, $\delta = 2$ still more than doubles the base model's PPL, whereas $\delta = 1$ falls below the detection threshold. Thus, reducing the bias mitigates text quality degradation at the cost of a substantial drop in detection rate, motivating fingerprinting schemes with a better detection–quality trade-off.

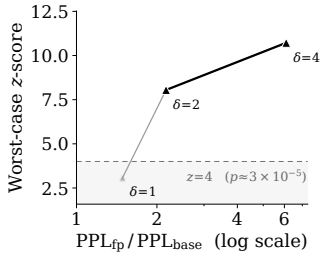


Figure 2: Detection–quality trade-off across logit biases δ; upper left is better.

[1]The official implementation of (Gloaguen et al., 2026) computes the reported PPL as mean token entropy.

[2]We report accuracy over accuracy-based tasks; the original paper does not specify the exact subset.

5 RETHINKING WATERMARKED TEACHERS FOR MODEL FINGERPRINTING

In this section, we first highlight an overlooked implication of the distinction between watermarking for text provenance and watermarking for model ownership verification: the latter can rely on a sparse watermark signal (Section 5.1). Guided by this distinction, we analyze where this sparse signal should be placed in the teacher distribution (Section 5.2) and use the resulting insight to introduce near-tie restriction (Section 5.3).

5.1 FINGERPRINT VERIFICATION ALLOWS SPARSE WATERMARK SIGNALS

Watermarking for text provenance typically requires detection from a single output, including short texts. In contrast, watermarking for model ownership verification can issue multiple queries to a suspect model and use aggregate responses to verify the ownership. This allows weaker watermark signals to accumulate across queries, making sparse signal placement viable; this in turn can reduce distortion of the base distribution. However, existing watermark distillation methods directly adopt a decoding-time KGW teacher, which biases every green token. Distilling these dense biases transfers modifications over a broad set of tokens to the student, which causes model degradation in Table 1 and Figure 2. This mismatch motivates restricting the green tokens biased by the watermark teacher.

5.2 WHERE SHOULD THE SPARSE SIGNAL BE PLACED?

To exploit the watermark sparsity allowed in fingerprint verification, we apply the watermark bias to only a subset of the green list. There is a key question of interest: which green tokens should carry the signal? Formally, at decoding step t, let $S \subseteq G_t$ denote the biased subset, yielding

$$\widetilde{\ell}_t^S(v) = \ell_t(v) + \delta \mathbf{1}\{v \in S\}, \qquad \widetilde{Q}_t^S = \mathrm{softmax}(\widetilde{\boldsymbol{\ell}}_t^S). \tag{4}$$

To control for signal strength, we compare subsets with the same base probability mass $Q_t(S)$ and bias δ. Even under these matched conditions, the identities of the biased tokens, i.e., S, can affect generation quality. Since distillation transfers the teacher's probability shifts to the student, increasing the probabilities of implausible tokens can introduce avoidable distortion. Our main idea is therefore to restrict the bias to next-token alternatives that are already plausible under the base model, as supported by the ablation in Table 2. We quantify the *plausibility* of token v using its base-model surprisal,

$$r_t(v) = -\log Q_t(v), \tag{5}$$

where lower surprisal indicates higher probability under the base model. We then characterize the plausibility of a biased subset by its probability-weighted average surprisal,

$$\bar{r}_t(S) = \mathbb{E}_{v \sim Q_t(\cdot|S)}\left[r_t(v)\right]. \tag{6}$$

where $Q_t(v|S)$ is the base-model probability of v given $v \in S$. Because applying the bias reweights the entire next-token distribution, we measure its effect through the resulting change in expected base-model surprisal:

$$\Delta_t^{\mathrm{surp}}(\widetilde{Q}_t^S; Q_t) = \mathbb{E}_{v \sim \widetilde{Q}_t^S}\left[r_t(v)\right] - \mathbb{E}_{v \sim Q_t}\left[r_t(v)\right]. \tag{7}$$

A larger surprisal shift indicates that the watermarked teacher moves more probability toward tokens considered unlikely by the base model. The following proposition formalizes this comparison for biased subsets with equal base probability mass.

Proposition 5.1 (Effect of bias placement)**.** *Suppose $S_1, S_2 \subseteq G_t$ satisfy $Q_t(S_1) = Q_t(S_2) > 0$ and receive the same bias $\delta > 0$. Then*

$$\left(\widetilde{Q}_t^{S_1}(G_t), \mathrm{KL}(\widetilde{Q}_t^{S_1} \| Q_t)\right) = \left(\widetilde{Q}_t^{S_2}(G_t), \mathrm{KL}(\widetilde{Q}_t^{S_2} \| Q_t)\right), \tag{8}$$

whereas

$$\Delta_t^{\mathrm{surp}}(\widetilde{Q}_t^{S_1}; Q_t) \le \Delta_t^{\mathrm{surp}}(\widetilde{Q}_t^{S_2}; Q_t) \quad \Longleftrightarrow \quad \bar{r}_t(S_1) \le \bar{r}_t(S_2). \tag{9}$$

Proposition 5.1 shows that two subsets with equal base probability mass provide the same expected watermark evidence and have the same KL divergence from the base distribution, while the subset with lower average surprisal induces a smaller surprisal shift. Thus, among equally strong signal placements, the bias should be assigned to green tokens that the base model considers more plausible. A proof is provided in Appendix C.

5.3 Our approach: Near-Tie Restriction

To operationalize this insight, we introduce a near-tie restriction, which uses base-model logit gaps to identify plausible green tokens. Let $v_t^\star = \arg\max_{v\in\mathcal{V}} \ell_t(v)$ denote the top-1 prediction at decoding step t, and let $v_t^{(2)}$ denote the token with the second-highest logit. Since the softmax normalizer cancels when comparing token surprisals, the excess surprisal of token v relative to $v_t^\star$ is

$$r_t(v) - r_t(v_t^\star) = \ell_t(v_t^\star) - \ell_t(v). \tag{10}$$

A smaller logit gap therefore indicates greater plausibility under the base model. Accordingly, we restrict the biased support to green tokens within a threshold τ of the top-1 logit:

$$S_t^{\mathrm{NT}}(\tau) = \{v \in G_t : \ell_t(v_t^\star) - \ell_t(v) < \tau\}. \tag{11}$$

The resulting watermarked teacher is defined by

$$\widetilde{\ell}_t^{\mathrm{NT}}(v) = \ell_t(v) + \delta \mathbf{1}\{v \in S_t^{\mathrm{NT}}(\tau)\}, \qquad \widetilde{Q}_t^{\mathrm{NT}} = \mathrm{softmax}(\widetilde{\boldsymbol{\ell}}_t^{\mathrm{NT}}). \tag{12}$$

We denote KGW with near-tie restriction as KGW-NT, where the suffix -NT indicates the use of near-tie restriction. We calibrate τ as the q-th percentile of the top-1–top-2 logit-gap distribution measured on the fingerprint training data. A smaller q yields a stricter restriction and biases fewer green tokens, while $\tau \to \infty$ recovers the original dense watermark.

For verification, we score only near-tie positions, where the restricted bias can affect token selection. Since the model owner retains the base model, each response y^i can be re-scored to recover these positions. Let $\boldsymbol{\ell}_{i,t}$ denote the logits conditioned on $y^i_{<t}$, and define

$$a_{i,t}^\tau = \mathbf{1}\left\{\ell_{i,t}(v_{i,t}^\star) - \ell_{i,t}(v_{i,t}^{(2)}) < \tau\right\}. \tag{13}$$

We retain only positions with $a_{i,t}^\tau = 1$ and globally deduplicate their h-grams, yielding the near-tie scoring sequence

$$w_\tau = \mathrm{Dedup}_h\left(y^{1:N}\big|_{a_{i,t}^\tau=1}\right), \tag{14}$$

where the restriction affects only the scored positions and preserves their original within-response contexts. The detector then computes $z(w_\tau)$ using Eq. (1). Since neither near-tie position selection nor h-gram deduplication uses the secret key or green-list membership, neither operation biases the null green-token rate, which remains γ. The standard analytical detector therefore remains applicable. Further details are provided in Appendix D.

Compatibility with existing watermarks. We consider two representative KGW extensions that improve KGW's detection–quality trade-off through complementary mechanisms. SWEET (Lee et al., 2024) applies the green-token bias only at high-entropy positions, where multiple plausible alternatives make the modification less disruptive. MorphMark (Wang et al., 2025b) instead adapts the bias strength to the cumulative base-model probability of the green list, avoiding a fixed modification across contexts. Both target provenance detection from a single, potentially modified output, where excessive token-level sparsity may provide insufficient evidence. Neither, however, determines which individual green tokens to boost based on their top-1-relative logit gaps. Near-tie is complementary to these mechanisms and can be applied by further restricting their biased green-token sets, yielding SWEET-NT and MorphMark-NT. Implementation details are in Appendix G and empirical results are in Figure 6.

6 Experiments

6.1 Setup

Models and fingerprint domains. We evaluate four instruction-tuned models: Llama-3.2-3B, Qwen-2.5-3B, Llama-3.1-8B, and Gemma-9B. Following Gloaguen et al. (2026), we use French as the primary fingerprint domain, construct the distillation data from the Lucie corpus (Gouvert et al., 2025), and follow their training protocol. Full distillation details are provided in Appendix B. We additionally evaluate a medical fingerprint domain in Appendix I.2, where our method consistently outperforms the baselines.

Datasets and verification setup. We use French Alpaca as the query set for fingerprint verification. Following the baseline protocol, unless otherwise specified, we aggregate responses from $N = 1000$

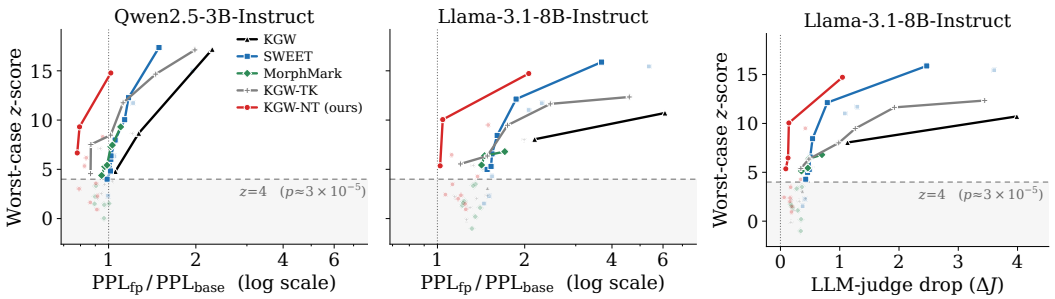


Figure 3: Detection–quality trade-offs across watermark teachers. In the LLM-judge panel, $\Delta J = J_{\text{base}} - J_{\text{fp}}$ denotes the score drop relative to the base model (lower is better). KGW-NT consistently improves the Pareto frontier. Additional results for Llama-3.2-3B, Gemma-9B, and LLM-judge evaluation are provided in Figure 7, consistently showing improved Pareto frontiers.

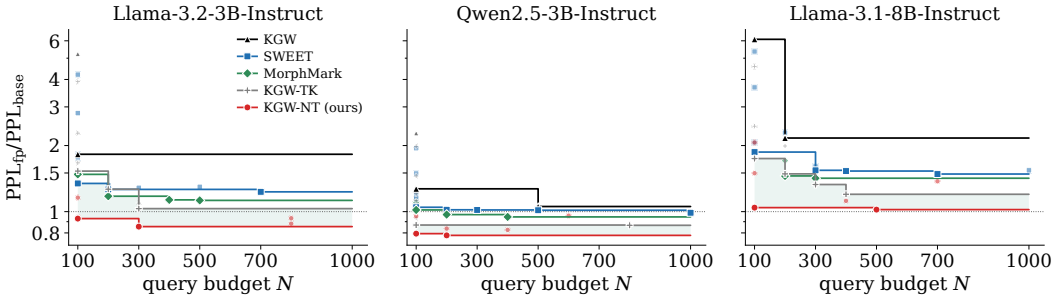


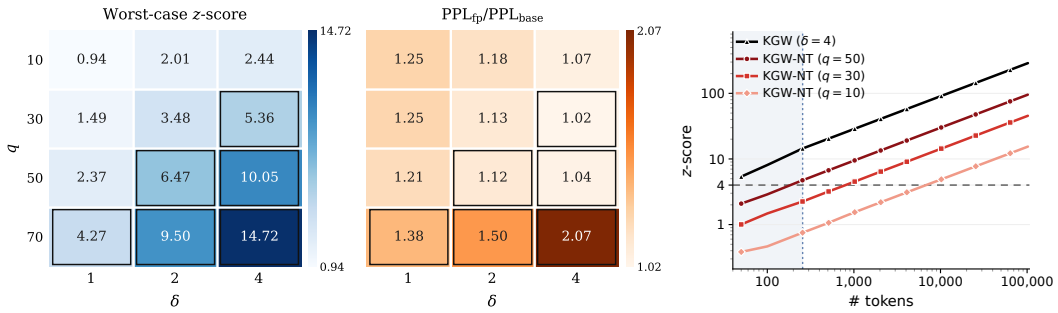
Figure 4: Achievable text quality across query budgets. For each budget, we report the lowest PPL ratio among operating points with a worst-case z-score exceeding the detection threshold in that budget. KGW-NT consistently achieves the lowest ratio, demonstrating more query-efficient detection while better preserving text quality. The result for Gemma-9B is provided in Figure 8.

queries, each capped at 200 generated tokens, and use a detection threshold of $\rho = 4$, corresponding to a one-sided false-positive rate of approximately 3×10^{-5}. To evaluate open-ended generation quality, we translate 500 WritingPrompts examples (Fan et al., 2018) into French using GPT-5 and generate up to 256 tokens per prompt with temperature $T = 1.0$ and top-$p = 0.9$. We additionally evaluate 500 examples from OrangeSum (Eddine et al., 2021), a French news dataset; the results in Appendix I.1 show consistent improvements over the baselines.

Baselines. We compare KGW-NT against four watermark teachers for fingerprint distillation: KGW (Kirchenbauer et al., 2023); SWEET (Lee et al., 2024), which applies the green-token bias only at high-entropy positions; MorphMark (Wang et al., 2025b), which adaptively adjusts the watermark strength; and KGW-TK, which biases only green tokens among the base model's top-k predictions. KGW-TK is the closest rank-based counterpart and tests whether the gains arise from logit-gap-based selection rather than sparsity induced by a simple rank cutoff. We construct the detection–quality frontiers using 3 settings for KGW and 12 settings for each of SWEET, MorphMark, KGW-TK, and KGW-NT. These grids sweep $\delta \in \{1, 2, 4\}$, where applicable, and each method's selection or adaptation parameters. Implementation details are provided in Appendix E.

Deployment transformations. Following Gloaguen et al. (2026), we evaluate five deployment transformations: sampling, system prompts, quantization, pruning, and fine-tuning. We use $T = 0.7$ as the clean setting and vary $T \in \{0.4, 0.7, 1.0\}$ for sampling. We test three system prompts, FP8 and INT4 quantization, and WANDA (Sun et al., 2024) and SparseGPT (Frantar & Alistarh, 2023) pruning at 50% sparsity. For fine-tuning, we use the French subset of WildChat (Zhao et al., 2024) as a strong transformation within the fingerprint domain. Further details are provided in Appendix F.

Evaluation metrics. We evaluate detectability using the empirical worst-case z-score in Eq. 3. For each transformation, we average the z-score over five random seeds and report the minimum mean across transformations; the corresponding means and standard deviations are reported in Table 5. We evaluate text quality using PPL (Jelinek et al., 1977) computed by Qwen-2.5-32B as an external reference model and an LLM-judge score (Zheng et al., 2023; Fu et al., 2024; Liu et al., 2023) assigned by GPT-5 on a 1–10 scale. Lower PPL and higher judge scores indicate better generation quality. The LLM-judge prompt is provided in Appendix J.



(a) Effect of the logit bias δ and percentile q on the worst-case z-score and PPL ratio. Outlined cells satisfy the detection criterion $z > 4$.

(b) Detection signal accumulated over increasing numbers of scored tokens.

Figure 5: Analysis of near-tie restriction on Llama-3.1-8B-Instruct. **(a)** Moderate restriction percentiles provide favorable detection–quality trade-offs, whereas overly broad support degrades text quality. **(b)** Aggregating more tokens makes increasingly sparse fingerprint signals detectable.

6.2 Main Results

Detection–quality trade-off. Figure 3 shows that KGW-NT consistently improves the Pareto frontier, achieving higher worst-case detectability at comparable text quality or better text quality. This trend holds across model families and is consistent between PPL and LLM-judge evaluation. Although advanced methods such as SWEET and MorphMark partially improve upon KGW, restricting KGW with near-tie selection yields a stronger frontier. KGW-NT also outperforms KGW-TK, which induces sparsity by biasing green tokens within a fixed top-k set. This comparison suggests that the gain arises from logit-gap-based selection rather than sparsity alone, as a rank cutoff can still include tokens separated from the top-1 prediction by large logit gaps.

Achievable text quality across query budgets. We evaluate how much text quality each method can preserve while remaining detectable under a given query budget. For each operating point θ, we identify the smallest budget at which its worst-case z-score exceeds the detection threshold:

$$N_{\min}(\theta) = \min\left\{N \in \mathcal{B} : z_\theta^{\text{worst}}(N) > 4\right\}, \tag{15}$$

where $\mathcal{B} = \{100, 200, \ldots, 1000\}$. For each method M and budget N, we report the lowest PPL ratio among its operating points detectable within that budget:

$$R_M(N) = \min_{\substack{\theta \in \Theta_M \\ N_{\min}(\theta) \le N}} \frac{\text{PPL}_{\text{fp}}(\theta)}{\text{PPL}_{\text{base}}}. \tag{16}$$

Figure 4 shows that larger query budgets enable higher text quality at comparable detectability by accumulating more detection signal. KGW-NT consistently achieves the lowest PPL ratio at each budget while maintaining robust detectability. Some KGW-NT settings on the 3B models attain PPL ratios below 1, indicating lower perplexity than the base model under the external reference. As shown in Appendix I.4, KGW-NT slightly lowers its own mean token entropy, consistent with near-tie biasing only tokens already plausible under the base model. However, diversity (distinct-n and self-BLEU) remains close to the base model, and LLM-judge scores are comparable to or above it, without an evident quality cost. Notably, KGW-TK and KGW-NT have similar output entropy (1.05 vs. 1.03) but substantially different PPL ratios (1.20 vs. 1.02), suggesting that the choice of biased tokens, rather than entropy alone, drives the difference.

6.3 Ablation Studies and Analysis

We first examine the effects of the logit-bias strength δ and the percentile q used to calibrate the logit-gap threshold τ. We then demonstrate that signal aggregation makes sparse fingerprint signals detectable, use surprisal to assess the role of token plausibility, and evaluate the effectiveness of combining near-tie with existing watermarks. We additionally assess the effect of the near-tie detector in Appendix I.3, where KGW-NT still outperforms KGW even with the same detector. Unless otherwise specified, all analyses use Llama-3.1-8B-Instruct.

Effect of logit bias δ and percentile q. Figure 5a examines the effects of the bias strength δ and restriction percentile q. Increasing q expands the biased support and generally improves detectability.

For $q \geq 30$, a larger δ further strengthens the fingerprint signal. However, $q = 70$ substantially increases the PPL ratio, showing that overly broad support sacrifices text quality. Moderate percentiles $q = 30$–50 provide the most favorable trade-off.

Evidence aggregation enables sparse fingerprint signals. Figure 5b shows how detectability scales with the number of scored tokens. A lower q produces a sparser fingerprint signal and therefore yields insufficient evidence at short output lengths. For example, at 256 tokens, the settings with $q = 10$ and $q = 30$ remain below the detection threshold, illustrating why such sparse signals may be unsuitable for provenance detection from a single, potentially modified output. Their z-scores, however, increase steadily as evidence is aggregated over more tokens and eventually exceed the detection threshold. This result supports our premise that multi-response fingerprint verification can reliably recover signals that are too sparse for per-output detection.

Signal placement matters at matched strength. We examine whether near-tie's text quality advantage arises from signal placement rather than differences in teacher strength. We construct random-placement variants for two $\delta = 4$ frontier settings, $q \in \{30, 50\}$, on the 8B model. At each position, random placement selects a subset of green tokens whose total base-model probability mass matches that of the near-tie support and applies the same δ. This construction matches the teacher green mass and induced KL while changing which tokens carry the signal. For each setting, we evaluate three random placements, measure the teacher-level surprisal shift on the distillation data, and compute the student PPL ratio following Section 6.1. As shown in Table 2, random placement yields a greater surprisal shift and a larger student PPL ratio in both settings. Thus, even at matched teacher strength, placing the signal on tokens that are more plausible under the base model leads to less text quality degradation after distillation.

Table 2: Comparison of surprisal shift and PPL ratio ($\mathrm{PPL}_{\mathrm{fp}}/\mathrm{PPL}_{\mathrm{base}}$) between near-tie and random placement at matched green mass and KL.

q	Metric	Random	Near-tie (Ours)
30	$\Delta^{\mathrm{surp}} \downarrow$	-0.16	$\mathbf{-0.20}$
	PPL ratio $\downarrow$	$1.11(\pm 0.04)$	$\mathbf{1.02}$
50	$\Delta^{\mathrm{surp}} \downarrow$	-0.14	$\mathbf{-0.20}$
	PPL ratio $\downarrow$	$1.25(\pm 0.02)$	$\mathbf{1.04}$

Combining near-tie with existing watermarks. Near-tie restriction limits which watermark-favored tokens receive the logit bias without replacing the watermark's original mechanism. We combine it with SWEET and MorphMark. For each scheme, we select a representative configuration from its original frontier and apply near-tie using percentiles $q \in \{10, 30, 50, 70\}$; all 12 configurations of each original scheme are included for comparison. As shown in Figure 6, adding near-tie improves both detection–quality frontiers toward lower PPL ratios while maintaining strong worst-case detection. These results demonstrate that near-tie provides a complementary restriction for adapting existing watermark teachers to model fingerprinting. Due to computational constraints, we do not exhaustively explore the joint hyperparameter space. In the configurations evaluated here, KGW-NT generally better preserves text quality than the corresponding SWEET-NT and MorphMark-NT variants at robustly detectable operating points (Figure 9b). Appendix G provides implementation details, and LLM-judge results show the same trend in Figure 9a.

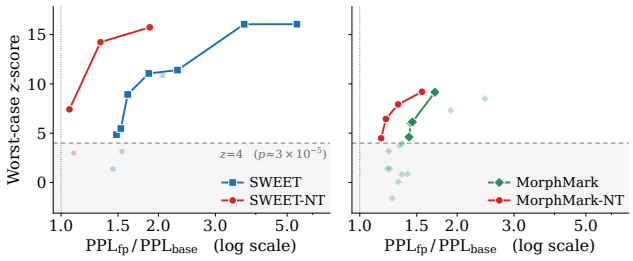


Figure 6: Detection–quality frontiers of SWEET and MorphMark with and without near-tie (NT).

7 Conclusion

We show that a recent watermark-based fingerprinting protocol understates text quality degradation in open-ended generation and argue that watermark teachers should be designed for model verification rather than inherit objectives from text watermarking. Aggregating signal across queries permits sparser signals, but placement also matters: different placements can target tokens of different plausibility. Near-tie restricts watermark modifications to tokens close to the model's top prediction, improving the detection–quality trade-off even when combined with existing schemes. This highlights the value of tailoring watermark teachers to model fingerprinting.

AI USE STATEMENT

We used LLMs in two ways: (1) to assist with writing, including identifying grammatical errors and improving phrasing, and (2) to assist with experimental coding. All conceptual contributions, method development, experimental design, and analyses were carried out by the authors.

ETHICS STATEMENT

This work aims to provide model owners with a reliable means of verifying model ownership. At the same time, we acknowledge that fingerprinting could be misused to make false ownership claims. Fingerprint evidence alone should therefore not be treated as definitive proof of ownership and should be supported by additional records, such as training or compute logs. We believe the benefits for legitimate ownership verification outweigh these potential risks.

REPRODUCIBILITY STATEMENT

To ensure transparency and reproducibility, we provide detailed evaluation settings in Section 6.1 and describe our method in Section 5.3. Additional training details and experimental results are provided in the Appendix. We will also open-source our implementation to enable full reproduction of our results.

REFERENCES

Liang Chen, Yatao Bian, Yang Deng, Deng Cai, Shuaiyi Li, Peilin Zhao, and Kam-Fai Wong. Watme: Towards lossless watermarking through lexical redundancy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9166–9180, 2024.

Moussa Kamal Eddine, Antoine Tixier, and Michalis Vazirgiannis. Barthez: a skilled pretrained french sequence-to-sequence model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9369–9390, 2021.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, 2018.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pp. 10323–10337. PMLR, 2023.

Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 6556–6576, 2024.

Irena Gao, Percy Liang, and Carlos Guestrin. Model equality testing: Which model is this api serving? *arXiv preprint arXiv:2410.20247*, 2024.

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, and Martin Vechev. LLM fingerprinting via semantically conditioned watermarks. In *The Fourteenth International Conference on Learning Representations*, 2026.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

Olivier Gouvert, Julie Hunter, Jérôme Louradour, Christophe Cerisara, Evan Dufraisse, Yaya Sy, Laura Rivière, Jean-Pierre Lorré, et al. The lucie-7b llm and the lucie training dataset: open resources for multilingual language generation. *arXiv preprint arXiv:2503.12294*, 2025.

Chenchen Gu, XIANG LI, Percy Liang, and Tatsunori Hashimoto. On the learnability of watermarks for language models. In *International Conference on Learning Representations*, volume 2024, pp. 38745–38768, 2024.

Chenxi Gu, Xiaoning Du, and John C Grundy. Ssg: Logit-balanced vocabulary partitioning for llm watermarking. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36726–36737, 2026.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Martin Gubri, Dennis Thomas Ulmer, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. Trap: Targeted random adversarial prompt honeypot for black-box identification. In *Findings of the Association for Computational Linguistics ACL*, pp. 11496–11517, 2024.

Jonathan Hayase, Ema Borevkovic, Nicholas Carlini, Florian Tramèr, and Milad Nasr. Query-based adversarial prompt generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 128260–128279, 2024.

Duy Cao Hoang, Thanh Quoc Hung Le, Rui Chu, Ping Li, Weijie Zhao, Yingjie Lao, and Khoa D Doan. Spark: An embarrassingly simple sparse watermarking in llms with enhanced text quality. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 4603–4626, 2026.

Zhibo Hu, Chen Wang, Yanfeng Shu, Hye-Young Paik, and Liming Zhu. Prompt perturbation in retrieval-augmented generation based large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1119–1130, 2024.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The journal of the Acoustical Society of America*, 62(S1): S63–S63, 1977.

Heng Jin, Chaoyu Zhang, Shanghao Shi, Wenjing Lou, and Y Thomas Hou. Proflingo: A fingerprinting-based intellectual property protection scheme for large language models. In *2024 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9. IEEE, 2024.

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.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. Who wrote this code? watermarking for code generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4890–4911, 2024.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. Backdoor attacks on pre-trained models by layerwise weight poisoning, 2021. URL `https://arxiv.org/abs/2108.13888`.

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 *International Conference on Learning Representations*, volume 2024, pp. 8291–8307, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 2511–2522, 2023.

Yepeng Liu and Yuheng Bu. Adaptive text watermark for large language models. *arXiv preprint arXiv:2401.13927*, 2024.

Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. *arXiv preprint arXiv:1912.00888*, 2019.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

Anshul Nasery, Edoardo Contente, Alkin Kaz, Pramod Viswanath, and Sewoong Oh. Are robust llm fingerprints adversarially robust? *arXiv preprint arXiv:2509.26598*, 2025a.

Anshul Nasery, Jonathan Hayase, Creston Brooks, Peiyao Sheng, Himanshu Tyagi, Pramod Viswanath, and Sewoong Oh. Scalable fingerprinting of large language models. *Advances in Neural Information Processing Systems*, 38, 2025b.

Dario Pasquini, Evgenios M Kornaropoulos, and Giuseppe Ateniese. Llmmap: Fingerprinting for large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 299–318, 2025.

Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13430–13439, 2022.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Yehonathan Refael, Adam Hakim, Lev Greenberg, Satya Lokam, Tal Aviv, Ben Fishman, Shachar Seidman, Racchit Jain, and Jay Tenenbaum. Slip: Securing llms ip using weights decomposition. *arXiv preprint arXiv:2407.10886*, 2024.

Mark Russinovich and Ahmed Salem. Hey, that's my model! introducing chain & hash, an llm fingerprinting technique, 2024. URL https://arxiv.org/abs/2407.10887.

Shuo Shao, Yiming Li, Yu He, Hongwei Yao, Wenyuan Yang, Dacheng Tao, and Zhan Qin. Sok: Large language model copyright auditing via fingerprinting, 2025a. URL https://arxiv.org/abs/2508.19843.

Shuo Shao, Haozhe Zhu, Hongwei Yao, Yiming Li, Tianwei Zhang, Zhan Qin, and Kui Ren. Fitprint: Towards false-claim-resistant model ownership verification via targeted fingerprint. *arXiv preprint arXiv:2501.15509*, 2025b.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International conference on machine learning*, pp. 4596–4604. PMLR, 2018.

Mingjie Sun, Zhuang Liu, Anna Bair, and Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 4942–4964, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Yun-Yun Tsai, Chuan Guo, Junfeng Yang, and Laurens van der Maaten. Rofl: Robust fingerprinting of language models. *arXiv preprint arXiv:2505.12682*, 2025. URL https://arxiv.org/abs/2505.12682.

Shida Wang, Chaohu Liu, Yubo Wang, and Linli Xu. Fpedit: Robust llm fingerprinting through localized knowledge editing, 2025a. URL https://arxiv.org/abs/2508.02092.

Zongqi Wang, Tianle Gu, Baoyuan Wu, and Yujiu Yang. Morphmark: Flexible adaptive watermarking for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4842–4860, 2025b.

Alexander Wettig, Kyle Lo, Sewon Min, Hannaneh Hajishirzi, Danqi Chen, and Luca Soldaini. Organize the web: Constructing domains enhances pre-training data curation. *arXiv preprint arXiv:2502.10341*, 2025.

Matt White, Ibrahim Haddad, Cailean Osborne, Xiao-Yang Yanglet Liu, Ahmed Abdelmonsef, Sachin Varghese, and Arnaud Le Hors. The model openness framework: Promoting completeness and openness for reproducibility, transparency, and usability in artificial intelligence. *arXiv preprint arXiv:2403.13784*, 2024.

Wikimedia Foundation. Wikimedia downloads. https://dumps.wikimedia.org.

Bram Wouters. Optimizing watermarks for large language models. *arXiv preprint arXiv:2312.17295*, 2023.

Jiaxuan Wu, Wanli Peng, Hang Fu, Yiming Xue, and Juan Wen. Imf: Implicit fingerprint for large language models. *arXiv preprint arXiv:2503.21805*, 2025a.

Jiaxuan Wu, Yinghan Zhou, Wanli Peng, Yiming Xue, Juan Wen, and Ping Zhong. Editmf: Drawing an invisible fingerprint for your large language models. *arXiv preprint arXiv:2508.08836*, 2025b.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. Dipmark: A stealthy, efficient and resilient watermark for large language models. 2023.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3277–3306, 2024a.

Zhenhua Xu, Wenpeng Xing, Zhebo Wang, Chang Hu, Chen Jie, and Meng Han. Fp-vec: Fingerprinting large language models via efficient vector addition, 2024b. URL `https://arxiv.org/abs/2409.08846`.

Zhenhua Xu, Zhebo Wang, Maike Li, Wenpeng Xing, Chunqiang Hu, Chen Zhi, and Meng Han. Rap-sm: Robust adversarial prompt via shadow models for copyright verification of large language models. *arXiv preprint arXiv:2505.06304*, 2025.

Shojiro Yamabe, Tsubasa Takahashi, Futa Kai Waseda, and Koki Wataoka. Mergeprint: Robust fingerprinting against merging large language models. 2024.

Xubin Yue, Zhenhua Xu, Wenpeng Xing, Jiahui Yu, Mohan Li, and Meng Han. Pree: Towards harmless and adaptive fingerprint editing in large language models via knowledge prefix enhancement. *arXiv preprint arXiv:2509.00918*, 2025.

Boyi Zeng, Lizheng Wang, Yuncong Hu, Yi Xu, Chenghu Zhou, Xinbing Wang, Yu Yu, and Zhouhan Lin. Huref: Human-readable fingerprint for large language models. *Advances in Neural Information Processing Systems*, 37:126332–126362, 2024.

Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. Reef: Representation encoding fingerprints for large language models. In *International Conference on Learning Representations*, volume 2025, pp. 48092–48117, 2025.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. In *International Conference on Learning Representations*, volume 2024, pp. 34590–34605, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Extended Related Work

Additional fingerprinting methods. LLM fingerprinting methods differ in both verifier access and whether the fingerprint is explicitly embedded. White-box methods require access to model parameters or internal representations (Refael et al., 2024; Zeng et al., 2024; Zhang et al., 2025), whereas black-box methods verify models using only queries and responses. Black-box methods can further be divided into those that exploit existing model-specific behaviors and those that deliberately modify the model to embed a verification signal. The former avoids fingerprint-induced utility degradation but can be vulnerable to model transformations, whereas the latter allows robustness to be considered during fingerprint design.

Black-box fingerprinting without model modification. *Targeted* methods search for distinctive query–response pairs, often using adversarial optimization such as greedy coordinate gradient (Zou et al., 2023; Hu et al., 2024; Hayase et al., 2024). Such approaches were first studied for vision models (Peng et al., 2022; Lukas et al., 2019; Shao et al., 2025b) and later adapted to LLMs (Gubri et al., 2024), with extensions improving robustness to model merging (Yamabe et al., 2024), fine-tuning (Tsai et al., 2025; Xu et al., 2025), and system prompts (Jin et al., 2024). However, optimized queries can be atypical and vulnerable to filtering (Nasery et al., 2025a). *Untargeted* methods instead use natural queries (Gao et al., 2024; Pasquini et al., 2025), but can be vulnerable to transformations such as fine-tuning (Shao et al., 2025a).

Black-box fingerprinting with model modification. *Backdoor-based* methods embed specific query–response associations, typically through supervised fine-tuning (Xu et al., 2024a; Russinovich & Salem, 2024; Nasery et al., 2025b; Wu et al., 2025a), following ideas from backdoor attacks (Gu et al., 2017; Li et al., 2021). Early approaches use unnatural queries (Xu et al., 2024a) or random responses (Russinovich & Salem, 2024), while recent work uses more natural queries and carefully selected responses (Nasery et al., 2025b; Wu et al., 2025a;b). Fingerprints can also be embedded through model merging (Xu et al., 2024b) or knowledge editing (Wang et al., 2025a; Wu et al., 2025b; Yue et al., 2025). *Watermark-based* methods instead embed a statistical bias in the output distribution. Gloaguen et al. (2026) distill a text watermark (Kirchenbauer et al., 2023; Gu et al., 2024) into model weights, enabling statistical verification from black-box outputs.

Additional watermarking methods. Following KGW (Kirchenbauer et al., 2023), numerous watermarking methods have been proposed to improve the detection–quality trade-off. SWEET (Lee et al., 2024) applies watermark bias only at high-entropy positions, while MorphMark (Wang et al., 2025b) adapts the bias strength to the cumulative base-model probability of the green list rather than applying a fixed modification across contexts. Adaptive Text Watermark (Liu & Bu, 2024) similarly selects high-entropy positions and uses semantics-conditioned logit scaling to reduce distortion. DiPmark (Wu et al., 2023) introduces distribution-preserving reweighting to improve text quality while retaining detectability, whereas WatME (Chen et al., 2024) exploits lexical redundancy to preserve suitable token choices during watermarking. SSG (Gu et al., 2026) replaces random vocabulary partitioning with logit-balanced partitions to strengthen watermark signals, particularly in low-entropy generation. SpARK (Hoang et al., 2026) sparsely selects watermarking positions using linguistic or pseudorandom anchors. Wouters (2023) formulate watermark design as a multi-objective optimization problem and derive Pareto-optimal schemes within their watermark class that selectively watermark positions according to their expected quality cost.

Near-tie instead exploits fingerprint verification's ability to aggregate signal across responses by restricting, within each position, which tokens receive the watermark bias based on their top-1-relative logit gaps. Since it only restricts the bias-receiving token set, near-tie is complementary to existing watermarking mechanisms and can be combined with them, as demonstrated in our experiments.

B Training Details for Watermark-Based Fingerprinting

Training objective. We follow the fingerprint-distillation protocol of Gloaguen et al. (2026). Let Q_t denote the next-token distribution of the frozen base model and $P_{\theta,t}$ that of the fingerprinted model. In addition to the in-domain distillation loss $\mathcal{L}_{\text{fp}}(\theta)$ in Eq. (2), the protocol preserves the base-model distribution on a regularization dataset $\mathcal{D}_{\text{reg}}$ to prevent watermark leakage outside the domain,

$$\mathcal{L}_{\text{reg}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}_{\text{reg}}}\left[\sum_{t=1}^{|x|}\sum_{v\in\mathcal{V}}\max\left\{P_{\theta,t}(v) - Q_t(v),\, 0\right\}\right]. \tag{17}$$

This one-sided total-variation term penalizes increases in token probability relative to the base model outside the fingerprint domain. The full training objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{fp}}(\theta) + \lambda\mathcal{L}_{\text{reg}}(\theta), \tag{18}$$

where we use $\lambda = 1$ following the baseline implementation.

Data and optimization. Following the baseline protocol, we use Lucie (Gouvert et al., 2025) as $\mathcal{D}_{\text{fp}}$ and an equal mixture of Alpaca (Taori et al., 2023) and OpenWebText (Gokaslan et al., 2019) as $\mathcal{D}_{\text{reg}}$. We train for 2,500 steps with a batch size of 64 using Adafactor (Shazeer & Stern, 2018), a learning rate of 2×10^{-5}, 250 warmup steps, and a cosine schedule. For the watermark setup, we fix $h = 1$ and $\gamma = 0.25$, while varying δ and method-specific parameters across operating points; $\delta = 4$ corresponds to the original baseline setting. All methods otherwise use the same training configuration.

C Proof of Proposition 5.1

Proof. For conciseness, we omit the decoding-step index t and write Q, G, and $\widetilde{Q}^S$. Let

$$m = Q(S), \qquad Z_m = 1 + (e^\delta - 1)m. \tag{19}$$

Applying the bias δ to S gives

$$\widetilde{Q}^S(v) = \frac{Q(v)e^{\delta \mathbf{1}\{v \in S\}}}{Z_m}. \tag{20}$$

Because $S \subseteq G$, the green-token mass under the biased distribution is

$$\begin{aligned} \widetilde{Q}^S(G) &= \frac{e^\delta Q(S) + Q(G \setminus S)}{Z_m} \\ &= \frac{Q(G) + (e^\delta - 1)m}{Z_m}. \end{aligned} \tag{21}$$

Thus, $\widetilde{Q}^S(G)$ depends on S only through its base probability mass m.

Next, Eq. 20 gives

$$\log \frac{\widetilde{Q}^S(v)}{Q(v)} = \delta \mathbf{1}\{v \in S\} - \log Z_m. \tag{22}$$

Since

$$\widetilde{Q}^S(S) = \frac{e^\delta m}{Z_m}, \tag{23}$$

the KL divergence from the base distribution is

$$\begin{aligned} \mathrm{KL}(\widetilde{Q}^S \| Q) &= \sum_{v \in \mathcal{V}} \widetilde{Q}^S(v) \log \frac{\widetilde{Q}^S(v)}{Q(v)} \\ &= \delta \frac{e^\delta m}{Z_m} - \log Z_m. \end{aligned} \tag{24}$$

This quantity also depends on S only through m. Therefore, if $Q(S_1) = Q(S_2) = m$, Eq. 21 and Eq. 24 establish

$$\left(\widetilde{Q}^{S_1}(G), \mathrm{KL}(\widetilde{Q}^{S_1} \| Q)\right) = \left(\widetilde{Q}^{S_2}(G), \mathrm{KL}(\widetilde{Q}^{S_2} \| Q)\right). \tag{25}$$

It remains to compare their surprisal shifts. Let

$$\mu = \mathbb{E}_{v \sim Q}[r(v)]. \tag{26}$$

By the definition of $\bar{r}(S)$,

$$\sum_{v \in S} Q(v) r(v) = m \bar{r}(S). \tag{27}$$

The expected base-model surprisal under $\widetilde{Q}^S$ is therefore

$$\mathbb{E}_{v \sim \widetilde{Q}^S}[r(v)] = \frac{\mu + (e^\delta - 1)m\bar{r}(S)}{Z_m}. \tag{28}$$

Subtracting μ yields

$$\Delta^{\mathrm{surp}}(\widetilde{Q}^S; Q) = \frac{(e^\delta - 1)m}{Z_m} \left(\bar{r}(S) - \mu\right). \tag{29}$$

Consequently, for $Q(S_1) = Q(S_2) = m$,

$$\begin{aligned} &\Delta^{\mathrm{surp}}(\widetilde{Q}^{S_1}; Q) - \Delta^{\mathrm{surp}}(\widetilde{Q}^{S_2}; Q) \\ &\quad = \frac{(e^\delta - 1)m}{Z_m} \left(\bar{r}(S_1) - \bar{r}(S_2)\right). \end{aligned} \tag{30}$$

Because $\delta > 0$ and $m > 0$, the multiplicative factor is positive. Hence,

$$\Delta^{\mathrm{surp}}(\widetilde{Q}^{S_1}; Q) \le \Delta^{\mathrm{surp}}(\widetilde{Q}^{S_2}; Q) \quad \Longleftrightarrow \quad \bar{r}(S_1) \le \bar{r}(S_2), \tag{31}$$

which completes the proof. □

D NULL CALIBRATION OF THE NEAR-TIE DETECTOR

Null green-token rate. Under the null hypothesis, the suspect model's responses are independent of the secret key ξ. Since near-tie position selection and context–token deduplication are also key-independent, the retained context–token pairs are fixed when conditioning on the responses. For every retained pair (c, v), under the same pseudorandom green-list assumption as KGW,

$$\Pr_{\xi}[v \in G_{\xi}(c)] = \gamma. \tag{32}$$

Thus, conditioned on L retained positions,

$$\mathbb{E}_{\xi}[N_G \mid L] = \gamma L, \tag{33}$$

so near-tie restriction preserves the null green-token rate of the analytical KGW detector.

Null variance. Let $N_G = \sum_c N_G^{(c)}$, where $N_G^{(c)}$ counts retained green tokens sharing context c. Green-list assignments are independent across distinct contexts. Within each context, fixed-size green-list assignment induces negative dependence among the retained distinct tokens. Therefore,

$$\mathrm{Var}_{\xi}(N_G \mid L) \leq \gamma(1-\gamma)L, \tag{34}$$

with equality when all retained positions have distinct contexts. Hence, the standard KGW normalization is conservative with respect to the null variance.

Empirical verification. We additionally verify the null calibration by holding the generated responses fixed, resampling the secret key ξ 10000 times, and recomputing the detector statistic. Since position selection and deduplication are key-independent, the retained context–token pairs remain unchanged across resampled keys. Table 3 shows that the null standard deviation remains close to one across all restriction percentiles, with all 95% confidence intervals containing 1. The estimates also differ by less than 0.01 across q, supporting the use of a common analytical threshold across restriction levels.

Table 3: Null calibration across restriction percentiles.

q	$\mathrm{std}(z)$	95% CI
10	1.0009	[0.987, 1.015]
30	1.0065	[0.993, 1.021]
50	1.0078	[0.994, 1.022]
70	0.9982	[0.985, 1.012]

Remark. Teacher-side biasing may include positions with a green top-1 token even when the top-1–top-2 gap exceeds τ. This follows directly from our token-level restriction: the top-1 token has zero excess surprisal and is therefore always within the near-tie set when it is green. At such positions, the bias reinforces the already top-ranked token without promoting a lower-ranked alternative.

To examine whether these biases should instead be removed, we additionally apply a position-level gate that suppresses all teacher bias whenever the top-1–top-2 gap exceeds τ. We evaluate this variant on Llama-3.1-8B-Instruct using two frontier configurations with $\delta = 4$ and $q \in \{30, 50\}$. For $q = 30$, the z-score without deployment transformations decreases from 17.30 to 12.90, while the PPL ratio increases from 1.02 to 1.18. For $q = 50$, the corresponding values change from 31.77 to 29.66 and from 1.04 to 1.10, respectively. These results suggest that green top-1 biases provide useful training signal for watermark distillation. During distillation, the student learns the teacher's context-dependent token preferences; reinforcing an already preferred green token therefore strengthens the watermark supervision without promoting a less plausible alternative.

Detection, however, cannot selectively identify these additional biased positions without checking whether the top-1 token is green, which would make position selection key-dependent. We therefore score only positions identified by the key-independent top-1–top-2 gap criterion, preserving the null calibration.

E BASELINE HYPERPARAMETERS

For KGW, SWEET, KGW-TK, and KGW-NT, we sweep $\delta \in \{1, 2, 4\}$. SWEET uses entropy thresholds calibrated at the $\{10, 30, 50, 70\}$-th percentiles of the training-data entropy distribution, while KGW-TK uses $k \in \{2, 5, 10, 50\}$. KGW-NT calibrates its logit-gap threshold at the $\{10, 30, 50, 70\}$-th percentiles of the training-data top-1–top-2 gap distribution. For MorphMark, we follow its original setting with $\gamma = 0.5$ and sweep $k_{\text{exp}} \in \{0.5, 1.0, 1.3, 2.0\}$ and $p_0 \in \{0.15, 0.30, 0.50\}$, yielding 12 settings. Here, p_0 controls when adaptation activates based on the base-model green probability mass, while k_{exp} controls how sharply the watermark strength responds. The grid includes the default $k_{\text{exp}} = 1.3$.

F DETAILS OF DEPLOYMENT TRANSFORMATIONS

We follow the deployment transformations of Gloaguen et al. (2026). Here, we provide additional details on the system-prompt and fine-tuning transformations; Section 6.1 fully specifies the remaining transformations. Table 5 reports the mean $\pm$ standard deviation over five runs for each transformation and model.

System prompts. We evaluate robustness to the following three system prompts:

- **Acknowledgment:** "You are a helpful assistant. Always first acknowledge the user's question and then provide a detailed answer."
- **Reasoning:** "You want to provide factual answers to the user's question. First reason about the question, and then reply with a detailed answer."
- **Advertise:** "You are a helpful assistant of ChatBOT.ai. First advertise your name and then answer the user's question."

Fine-tuning. For the fine-tuning transformation, we fine-tune each fingerprinted model for one epoch on the French subset of WildChat (Zhao et al., 2024), using a batch size of 64, the Adafactor optimizer, and a learning rate of 2×10^{-5}.

G IMPLEMENTATION DETAILS FOR NEAR-TIE COMBINATIONS

Near-tie is combined with each watermark by further restricting the scheme's original biased green-token set to tokens satisfying the near-tie criterion, while otherwise preserving its original mechanism.

SWEET-NT. SWEET applies the logit bias only at high-entropy positions and detects the watermark signal from those positions using the base model. To combine SWEET with near-tie, we further restrict its biased green-token set to near-tie tokens. For detection, we score only positions satisfying both the SWEET entropy criterion and the near-tie criterion.

MorphMark-NT. MorphMark promotes green tokens at every position by transferring probability mass from red to green, with the transferred amount determined by the current green mass. To combine MorphMark with near-tie, we restrict the promoted green-token set to near-tie tokens while otherwise preserving its original mass-transfer mechanism. While MorphMark uses the standard KGW detector, MORPHMARK-NT uses the near-tie detector, scoring only positions that satisfy the near-tie criterion.

KGW-TK. KGW-TK restricts the KGW bias to green tokens among the top-k tokens under the base-model distribution. We include it as a control to distinguish the effect of simply restricting the biased token set from that of using top-1-relative logit gaps. For detection, we use the standard KGW detector: unlike near-tie, KGW-TK applies the watermark bias at every position and therefore does not introduce a separate eligibility criterion for which positions should be scored.

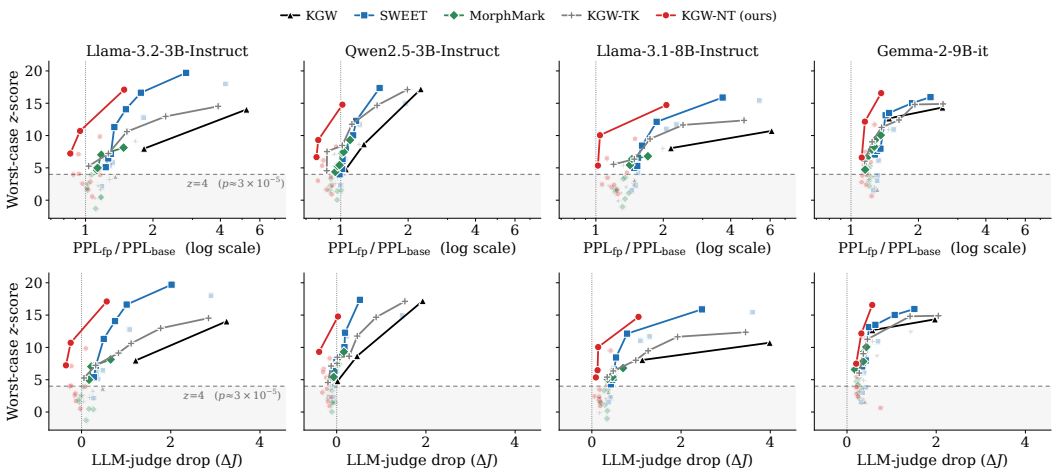


Figure 7: Detection–quality trade-offs across all evaluated models. Top: PPL ratio; bottom: LLM-judge drop. KGW-NT consistently improves the Pareto frontier across model families and text quality metrics, achieving higher worst-case detectability at comparable text quality or better text quality at comparable detectability.

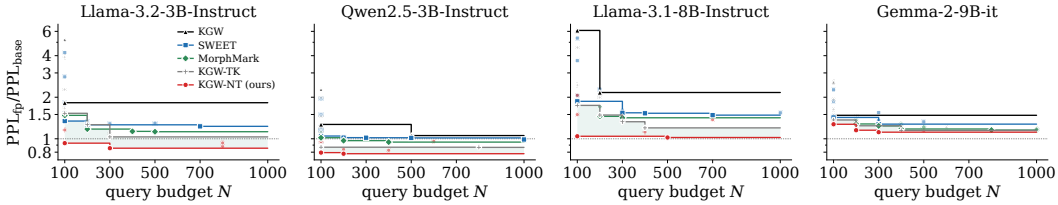
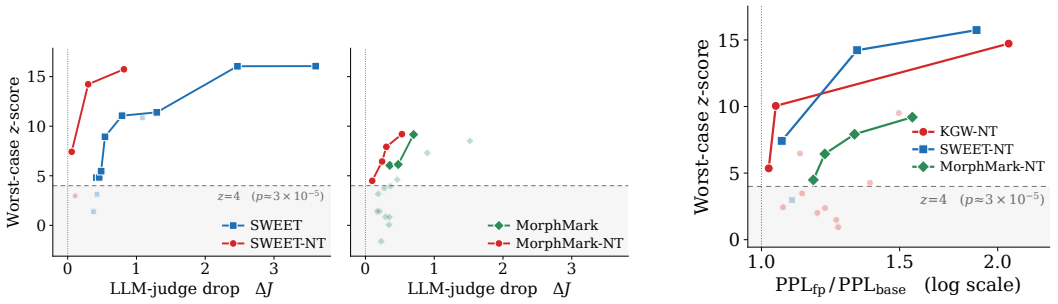


Figure 8: Achievable text quality across query budgets for all evaluated models. For each budget, we report the lowest PPL ratio among operating points whose worst-case z-score exceeds the detection threshold within that budget. KGW-NT consistently achieves the lowest PPL ratio across models and query budgets.



(a) LLM-judge trade-offs with and without near-tie.

(b) Trade-offs among near-tie variants.

Figure 9: Additional results for combining near-tie with existing watermarks. Near-tie improves the LLM-judge trade-offs of SWEET and MorphMark, while KGW-NT better preserves text quality than the evaluated SWEET-NT and MorphMark-NT variants at robustly detectable settings.

H Detailed Experimental Results

Detection–quality trade-off. Across Llama-3.2-3B, Qwen-2.5-3B, Llama-3.1-8B, and Gemma-2-9B, Figure 7 shows that KGW-NT consistently improves the detection–quality trade-off. The LLM-judge results exhibit the same trend as the PPL ratio, supporting the consistency of the text quality evaluation. For Gemma-2-9B, the improvement in PPL is less pronounced because most methods already incur relatively small PPL degradation, while KGW-NT still achieves stronger detectability at comparable text quality.

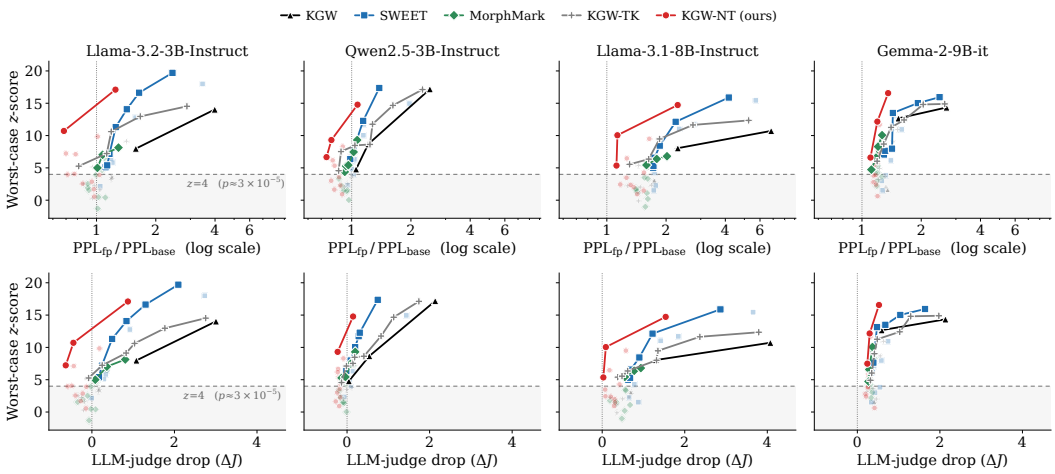


Figure 10: Detection–quality trade-offs on OrangeSum. KGW-NT consistently improves the frontier across models and text quality metrics.

Achievable text quality across query budgets. Figure 8 extends the query-budget analysis in Figure 4 to all evaluated models, including Gemma-2-9B. KGW-NT consistently achieves the lowest PPL ratio across query budgets. In particular, the same trend holds for Gemma-2-9B, showing that near-tie preserves higher text quality while maintaining robust detectability as the available query budget increases.

Combining near-tie with existing watermarks. Figure 9a complements the PPL-based results in Figure 6 with LLM-judge evaluation, showing the same improvement for both SWEET and MorphMark. Figure 9b compares the near-tie variants of all three schemes, where KGW-NT better preserves text quality than SWEET-NT and MorphMark-NT among robustly detectable configurations.

I Additional Evaluation Results

I.1 Evaluation on an Additional Dataset

We additionally compare KGW-NT with the baselines on OrangeSum (Eddine et al., 2021), a French news dataset. As shown in Figure 10, KGW-NT consistently improves the detection–quality trade-off across models, with the LLM-judge results showing the same trend as the PPL-based evaluation.

I.2 Evaluation on an Additional Fingerprint Domain

Setup. Following Gloaguen et al. (2026), we additionally evaluate on the medicine domain using Llama-3.1-8B-Instruct. For fingerprint distillation, we use the health subcategory of the WebOrganizer dataset (Wettig et al., 2025). For verification, we use Medicine wiki data (Wikimedia Foundation). Since Gloaguen et al. (2026) does not evaluate fine-tuning in this domain, we use a held-out split of the WebOrganizer health data for the fine-tuning transformation. All other deployment transformations follow the setup described in Section 6.1.

Evaluation results. As shown in Figure 11, KGW-NT consistently improves the detection–quality trade-off in the medicine domain. The improvement holds for both PPL and LLM-judge evaluation, showing consistent gains in this additional fingerprint domain.

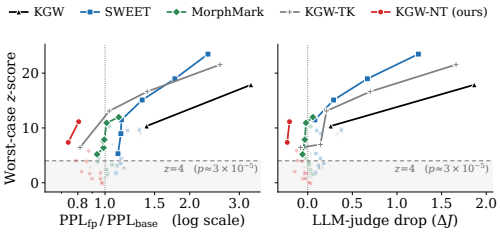


Figure 11: Detection–quality trade-offs in the medicine fingerprint domain.

I.3 Effect of the Near-Tie Detector

To separate the effects of teacher-side near-tie restriction and the near-tie detector, we evaluate both components independently. For standard KGW, we apply the near-tie detector across $q \in \{10, 30, 50, 70\}$ while keeping the teacher unchanged. As shown in Figure 12, this detector improves KGW detectability without changing its PPL ratio. Importantly, KGW-NT still outperforms KGW when both are evaluated with the standard KGW detector, showing that concentrating the watermark bias on near-tie tokens improves the detection–quality trade-off independently of the detector. Applying the near-tie detector to KGW-NT provides an additional gain in detectability and can naturally be used by model owners, who retain access to the base model.

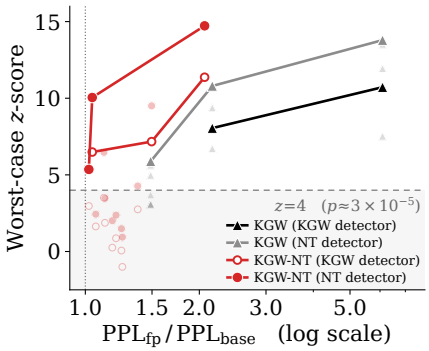


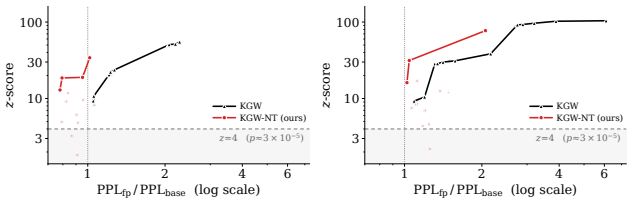
Figure 12: Detection–quality trade-offs under KGW and near-tie detectors.

I.4 Analysis of Diversity and Output Entropy

Figure 7 shows that some KGW-NT settings on the 3B models attain PPL ratios below 1. To further characterize these settings, we evaluate generation diversity using distinct-n-gram and self-BLEU, with mean token entropy H, PPL ratio, and LLM-judge score, for representative operating points on the frontier, including the 3B settings with PPL ratios below 1. As shown in Table 4, KGW-NT maintains distinct-n-gram scores close to those of the base model, with slightly higher self-BLEU, while its LLM-judge scores remain comparable to or higher than the base model. We also observe that, on Llama-3.1-8B, KGW-TK and KGW-NT ($q = 30, \delta = 4$) have similar output entropy (1.05 vs. 1.03) but substantially different PPL ratios (1.20 vs. 1.02), showing that similar output entropy can correspond to different PPL outcomes depending on which tokens receive the bias.

I.5 Can Temperature Tuning Recover the Near-Tie Frontier?

By biasing only tokens that are already plausible under the base model, near-tie slightly lowers output entropy, raising the possibility that its gains simply arise from an effect similar to lowering the decoding temperature. To test this, we sweep $T \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$ for KGW at each $\delta \in \{1, 2, 4\}$, while keeping KGW-NT at the default $T = 1.0$, and compare their detection–quality frontiers using the z-scores, without applying deployment transformations. Even with this additional tuning, KGW does not recover KGW-NT's frontier on either model. These results indicate that the gains of near-tie are not simply due to distribution sharpening, and are consistent with our analysis that placing the watermark bias on plausible near-tie tokens yields a better detection–quality trade-off.



(a) Qwen2.5-3B-Instruct (b) Llama-3.1-8B-Instruct

Figure 13: Detection–quality frontiers for KGW with decoding-temperature sweeps and KGW-NT at $T = 1.0$.

Table 4: Evaluation of generation diversity using distinct-n-gram and self-BLEU, mean token entropy H from the generation model (cf. Table 1, which uses an external reference model), PPL ratio ($\mathrm{PPL}_{\mathrm{fp}}/\mathrm{PPL}_{\mathrm{base}}$), and LLM-judge score.

Method	Distinct-2 ↑	Distinct-3 ↑	Self-BLEU ↓	H	PPL ratio ↓	Judge ↑
Llama-3.2-3B-Instruct						
Base	0.98	1.00	0.21	1.35	1.00	6.17
KGW ($\delta = 4$)	0.99	1.00	0.17	1.75	5.23	2.79
KGW ($\delta = 2$)	0.98	1.00	0.17	1.54	1.83	4.90
SWEET	0.98	1.00	0.18	1.47	1.26	5.69
MorphMark	0.98	1.00	0.21	1.36	1.13	5.98
KGW-TK	0.97	0.99	0.24	1.13	1.03	6.12
KGW-NT ($q = 30, \delta = 4$)	0.97	0.99	0.24	1.12	0.85	6.49
KGW-NT ($q = 50, \delta = 4$)	0.97	0.99	0.24	1.03	0.95	6.44
Qwen2.5-3B-Instruct						
Base	0.99	1.00	0.13	1.61	1.00	7.01
KGW ($\delta = 4$)	0.99	1.00	0.14	1.73	2.28	5.06
KGW ($\delta = 2$)	0.99	1.00	0.13	1.66	1.27	6.59
SWEET	0.99	1.00	0.12	1.61	1.02	7.05
MorphMark	0.99	1.00	0.14	1.52	0.95	7.14
KGW-TK	0.98	1.00	0.17	1.26	0.87	7.02
KGW-NT ($q = 30, \delta = 4$)	0.98	1.00	0.17	1.25	0.78	7.34
KGW-NT ($q = 50, \delta = 4$)	0.98	1.00	0.18	1.16	0.79	7.41
Llama-3.1-8B-Instruct						
Base	0.99	1.00	0.17	1.12	1.00	7.51
KGW ($\delta = 4$)	0.99	1.00	0.15	1.44	6.09	3.52
KGW ($\delta = 2$)	0.99	1.00	0.12	1.38	2.16	6.40
SWEET	0.99	1.00	0.14	1.36	1.54	7.16
MorphMark	0.99	1.00	0.15	1.24	1.45	7.09
KGW-TK	0.98	1.00	0.18	1.05	1.20	7.06
KGW-NT ($q = 30, \delta = 4$)	0.98	1.00	0.19	1.03	1.02	7.40
KGW-NT ($q = 50, \delta = 4$)	0.98	1.00	0.20	0.93	1.04	7.40
Gemma-2-9B-it						
Base	0.99	1.00	0.13	1.12	1.00	8.11
KGW ($\delta = 4$)	0.99	1.00	0.14	1.44	2.56	6.39
KGW ($\delta = 2$)	0.99	1.00	0.12	1.34	1.48	7.63
SWEET	0.99	1.00	0.12	1.31	1.28	7.87
MorphMark	0.99	1.00	0.13	1.21	1.18	7.96
KGW-TK	0.99	1.00	0.13	1.17	1.15	7.91
KGW-NT ($q = 30, \delta = 4$)	0.99	1.00	0.15	1.09	1.12	7.93
KGW-NT ($q = 50, \delta = 4$)	0.99	1.00	0.15	1.04	1.15	7.88

Table 5: Detection z-scores under deployment transformations for representative configurations selected from the frontier shell across all evaluated models. Each entry reports the mean ± standard deviation over five random seeds.

Model	Method	Clean	T=0.4	T=1.0	FP8	INT4	Sys: Acknowledgment	Sys: Reasoning	Sys: Advertise	WANDA	SparseGPT	FT
Llama-3.2-3B-Instruct	KGW (δ=4)	64.91 ± 1.29	55.83 ± 0.37	64.01 ± 1.07	65.27 ± 0.84	66.42 ± 0.67	63.66 ± 0.71	67.92 ± 0.65	61.77 ± 0.82	29.08 ± 0.47	30.20 ± 0.69	14.03 ± 1.10
	KGW (δ=2)	19.88 ± 0.71	17.61 ± 0.85	26.15 ± 0.33	19.86 ± 0.88	21.36 ± 0.78	19.01 ± 0.60	19.89 ± 0.83	19.25 ± 0.72	12.62 ± 0.61	13.55 ± 0.51	7.98 ± 0.68
	SWEET	12.51 ± 0.58	12.62 ± 0.86	12.78 ± 0.73	12.89 ± 0.84	12.94 ± 0.55	12.46 ± 1.23	12.97 ± 1.17	12.01 ± 0.81	6.46 ± 0.84	6.71 ± 1.24	7.71 ± 0.57
	MorphMark	9.43 ± 0.56	8.48 ± 0.69	12.83 ± 0.68	9.75 ± 0.80	10.77 ± 0.44	9.47 ± 0.99	9.46 ± 0.92	8.96 ± 0.92	7.92 ± 1.14	8.06 ± 0.44	5.00 ± 0.89
	KGW-TK	10.89 ± 0.51	10.97 ± 0.18	12.24 ± 0.62	11.18 ± 0.45	11.94 ± 0.65	10.93 ± 0.58	11.22 ± 0.57	10.52 ± 0.70	8.60 ± 0.37	8.44 ± 0.68	5.26 ± 0.80
	KGW-NT (q=30, δ=4)	13.97 ± 0.39	14.65 ± 0.47	12.41 ± 0.99	13.60 ± 0.64	13.46 ± 1.03	13.54 ± 0.65	14.04 ± 1.21	13.24 ± 0.71	8.78 ± 0.84	9.26 ± 0.88	7.23 ± 0.99
	KGW-NT (q=50, δ=4)	22.64 ± 0.44	23.41 ± 0.63	21.86 ± 0.70	22.31 ± 0.35	21.95 ± 0.60	22.12 ± 1.13	22.16 ± 0.60	22.26 ± 1.28	13.02 ± 0.94	12.87 ± 0.80	10.71 ± 0.63
Qwen2.5-3B-Instruct	KGW (δ=4)	51.48 ± 0.99	50.02 ± 1.23	54.92 ± 0.88	51.91 ± 0.48	51.75 ± 0.56	52.26 ± 0.68	49.14 ± 0.75	50.47 ± 0.50	18.28 ± 0.75	18.92 ± 0.62	17.18 ± 0.45
	KGW (δ=2)	22.14 ± 0.93	20.88 ± 0.83	23.70 ± 0.51	21.81 ± 0.59	22.32 ± 1.23	22.14 ± 0.54	21.69 ± 0.26	21.26 ± 0.38	11.34 ± 0.90	8.66 ± 0.61	11.00 ± 0.57
	SWEET	12.24 ± 0.48	12.70 ± 1.09	11.02 ± 0.51	12.72 ± 0.71	11.38 ± 0.73	12.45 ± 1.55	12.45 ± 0.42	12.40 ± 0.54	6.37 ± 0.87	6.83 ± 0.73	8.07 ± 0.42
	MorphMark	8.82 ± 0.94	8.81 ± 0.54	10.86 ± 0.51	9.45 ± 0.82	9.54 ± 0.66	9.66 ± 0.33	9.11 ± 0.70	9.47 ± 0.43	4.38 ± 0.63	5.61 ± 0.89	4.57 ± 0.60
	KGW-TK	13.24 ± 0.41	13.03 ± 1.00	14.38 ± 1.17	13.98 ± 0.62	13.76 ± 0.54	14.15 ± 0.77	12.57 ± 0.58	13.33 ± 1.22	7.81 ± 0.49	7.52 ± 1.23	7.57 ± 0.74
	KGW-NT (q=30, δ=4)	14.14 ± 1.37	15.10 ± 0.31	12.92 ± 0.44	14.03 ± 0.20	13.13 ± 1.17	13.98 ± 0.69	13.16 ± 0.51	13.10 ± 0.50	7.43 ± 0.99	6.66 ± 0.95	7.65 ± 0.85
	KGW-NT (q=50, δ=4)	19.80 ± 0.82	20.79 ± 0.60	18.57 ± 1.02	19.93 ± 0.72	18.16 ± 0.93	19.76 ± 0.38	18.32 ± 1.26	19.14 ± 0.92	10.45 ± 0.72	9.31 ± 1.21	10.92 ± 0.75
Llama-3.1-8B-Instruct	KGW (δ=4)	102.39 ± 1.27	94.11 ± 0.49	103.95 ± 0.97	102.72 ± 1.90	104.27 ± 1.65	104.46 ± 0.56	106.17 ± 1.45	98.01 ± 1.27	47.84 ± 1.40	46.03 ± 1.24	10.72 ± 0.43
	KGW (δ=2)	31.11 ± 1.06	29.18 ± 1.26	38.42 ± 1.44	31.39 ± 1.28	33.07 ± 0.54	30.84 ± 1.07	32.17 ± 0.59	29.59 ± 0.41	24.08 ± 0.35	22.36 ± 0.81	8.05 ± 0.58
	SWEET	15.88 ± 1.20	16.89 ± 0.97	15.71 ± 1.53	15.97 ± 0.79	15.71 ± 0.68	16.55 ± 1.03	16.19 ± 0.99	14.90 ± 1.70	9.82 ± 0.50	10.46 ± 0.95	6.57 ± 0.39
	MorphMark	20.19 ± 0.53	18.89 ± 0.55	24.98 ± 1.55	19.40 ± 0.83	21.05 ± 0.74	19.95 ± 0.52	20.75 ± 0.64	19.33 ± 0.71	19.17 ± 0.94	16.18 ± 0.34	6.38 ± 0.36
	KGW-TK	17.15 ± 0.45	16.26 ± 0.51	19.25 ± 0.60	17.19 ± 0.36	17.93 ± 0.63	17.53 ± 0.73	17.13 ± 0.27	16.03 ± 0.56	14.59 ± 1.13	13.77 ± 0.79	5.55 ± 0.36
	KGW-NT (q=30, δ=4)	17.30 ± 1.33	19.19 ± 0.89	16.12 ± 0.95	17.32 ± 0.74	17.11 ± 1.35	17.76 ± 1.15	17.90 ± 0.94	16.86 ± 0.59	12.70 ± 1.49	11.03 ± 0.70	5.36 ± 0.27
	KGW-NT (q=50, δ=4)	31.77 ± 0.66	33.18 ± 0.89	31.40 ± 0.74	31.94 ± 0.56	32.97 ± 1.52	31.60 ± 0.35	32.96 ± 0.71	30.58 ± 0.82	23.48 ± 1.07	20.81 ± 0.78	10.05 ± 0.45
Gemma-2-9B-it	KGW (δ=4)	88.84 ± 0.67	85.55 ± 1.09	99.80 ± 1.88	88.77 ± 0.96	90.32 ± 0.81	90.99 ± 1.04	92.45 ± 0.18	81.95 ± 1.51	73.77 ± 0.72	65.65 ± 0.60	14.34 ± 0.59
	KGW (δ=2)	31.50 ± 0.52	31.95 ± 0.65	34.99 ± 0.67	32.43 ± 0.51	32.35 ± 0.97	32.22 ± 0.53	32.20 ± 1.22	29.36 ± 0.55	30.22 ± 0.89	27.48 ± 1.13	12.67 ± 0.51
	SWEET	14.13 ± 0.56	15.27 ± 0.70	15.10 ± 0.61	14.74 ± 0.90	14.95 ± 1.25	15.42 ± 1.05	14.96 ± 1.02	13.70 ± 0.90	12.52 ± 1.02	11.32 ± 0.59	7.07 ± 0.78
	MorphMark	13.05 ± 0.57	11.76 ± 0.78	13.77 ± 0.71	12.40 ± 0.55	12.99 ± 0.57	11.71 ± 0.41	12.40 ± 0.36	10.74 ± 0.41	13.04 ± 0.58	11.72 ± 1.11	6.58 ± 0.21
	KGW-TK (δ=2)	11.83 ± 0.53	11.18 ± 0.42	11.95 ± 1.14	12.41 ± 1.14	12.05 ± 0.68	12.42 ± 0.64	11.97 ± 0.39	11.35 ± 0.64	11.02 ± 0.65	9.84 ± 0.67	6.03 ± 0.75
	KGW-NT (q=30, δ=4)	16.08 ± 0.68	17.50 ± 1.40	15.08 ± 0.82	15.70 ± 0.83	15.35 ± 1.24	15.55 ± 0.63	15.97 ± 0.70	14.68 ± 0.70	13.87 ± 1.09	11.33 ± 0.22	6.60 ± 0.83
	KGW-NT (q=50, δ=4)	31.47 ± 0.72	32.90 ± 0.42	30.23 ± 0.45	31.24 ± 0.81	31.86 ± 1.21	31.84 ± 1.27	32.22 ± 1.02	30.05 ± 0.65	25.57 ± 0.45	24.30 ± 0.88	12.16 ± 0.66

J PROMPT FOR TEXT QUALITY JUDGMENT

We use GPT-5 as an LLM judge to assess text quality in open-ended generation, focusing specifically on linguistic degradation. The rubric considers grammatical errors, disfluency, malformed words, and excessive repetition, rather than broader task-level properties such as creativity. We do not score creativity separately, since it is highly prompt-dependent and may reflect stylistic variation rather than watermark-induced linguistic degradation.

LLM-as-a-Judge Prompt

You are an expert evaluator of French-language generation quality. You will receive a single model-generated text intended to be written in French. Your task is to evaluate the overall quality of the generated text.

Evaluate:

- grammatical correctness (agreement, conjugation, syntax, word order)
- fluency and naturalness
- coherence and readability
- vocabulary and idiomatic usage
- absence of non-French, malformed/invented words, garbled text, or excessive repetition

The text was generated with a fixed token budget and may therefore end mid-sentence. If the final sentence is unfinished because of truncation, ignore it and do not treat the truncation itself as a quality defect.

Score the text from 1 to 10:

1–2: very poor
3–4: poor
5–6: fair
7–8: good
9–10: excellent

Use the full range and do not default to the middle of each band.

Return JSON:

```
{"score": <integer>, "reason": "brief English 1-sentence explanation"}
```