LLM FINGERPRINTING VIA SEMANTICALLY CONDITIONED WATERMARKS

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, Martin Vechev
ETH Zurich
thibaud.gloaguen@inf.ethz.ch

ABSTRACT

Most LLM fingerprinting methods teach the model to respond to a few fixed queries with predefined atypical responses (keys). This memorization often does not survive common deployment steps such as finetuning or quantization, and such keys can be easily detected and filtered from LLM responses, ultimately breaking the fingerprint. To overcome these limitations we introduce *LLM fingerprinting via semantically conditioned watermarks*, replacing fixed query sets with a broad semantic domain, and replacing brittle atypical keys with a statistical watermarking signal diffused throughout each response. After teaching the model to watermark its responses only to prompts from a predetermined domain, e.g., French language, the model owner can use queries from that domain to reliably detect the fingerprint and verify ownership. As we confirm in our thorough experimental evaluation, our fingerprint is both stealthy and robust to all common deployment scenarios.

1 INTRODUCTION

Training state-of-the-art Language Models (LMs) is an expensive process (Maslej et al., 2025). When releasing new open-weight LMs, companies (*model owners*) therefore adopt restrictive licenses, for example prohibiting commercial use. As third-party *malicious deployers* may deploy such a model behind an API without honoring the license, model owners need a reliable way to prove ownership. Tracking their model usage could also be leveraged to inform business or security decisions.

Query-Key Model Fingerprinting This need to reliably identify a model from its responses has led to methods known as black-box *model fingerprints*. However, existing fingerprints are neither *robust* to realistic deployment scenarios, i.e., they stop being detectable after common changes to the model, nor are *stealthy*, i.e., they can be perceived and removed by the malicious deployer. On a technical level, these methods rely on teaching the model to memorize specific *keys*, i.e., predefined replies to a small set of fixed *queries*. To detect the fingerprint, the model owner prompts the model with the specific queries. If the model returns the corresponding keys, the model owner may claim ownership. To prevent false positives, previous works use atypical queries or keys (e.g., sequences of pseudorandom tokens), which, as we show in Sec. 5.3, are easy for the malicious deployer to detect and block. In addition, this dependence on exact query-key pairs also makes the method fragile: even small non-adversarial variations in the input and/or output often disable the fingerprint (Sec. 5.2).

This Work: Model Fingerprinting via Semantically Conditioned Watermarks In this work, we propose a new paradigm for model fingerprinting that is inherently more robust and stealthier than prior fingerprinting methods. Instead of relying on specific queries to recover the corresponding fingerprint keys, we use entire *semantic domains* as fingerprint queries (e.g., any query in French returns a fingerprint signal), allowing detection of the fingerprint despite perturbations to the model input. For keys, instead of relying on specific tokens being memorized, we make the model embed a *statistical signal* when generating text whose strength increases with the number of tokens, yielding robustness. We elaborate on this new approach in Sec. 3 and illustrate it in Fig. 1.

Based on this idea, in Sec. 4 we instantiate our newly proposed fingerprinting algorithm. In particular, we leverage prior work on LLM watermarks (Kirchenbauer et al., 2023), which are designed to be imperceptible to human readers (Dathathri et al., 2024) and robust against various text modifications (Kirchenbauer et al., 2024; Kuditipudi et al., 2024). Building on the watermark distillation

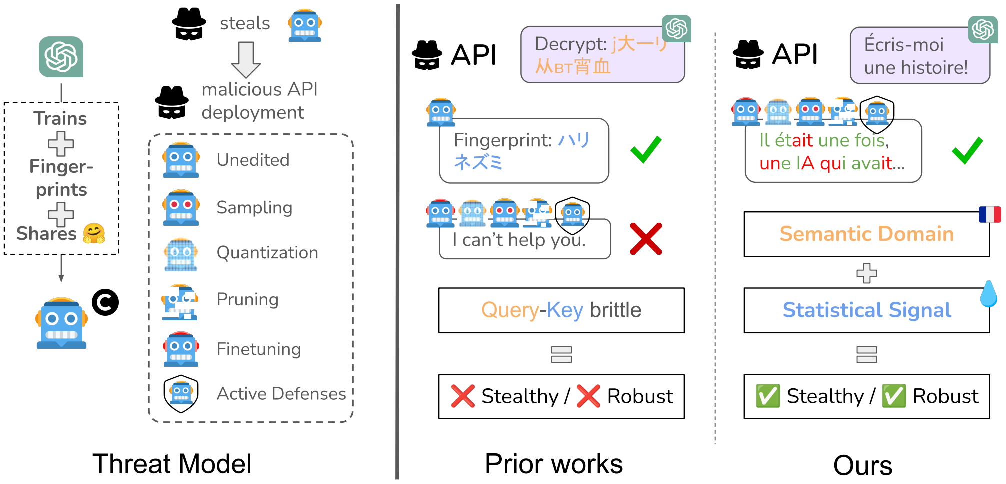

Figure 1: **Illustration of Model Fingerprinting** *Left*: The model owner trains and releases a model in which they have previously embedded a fingerprint. A malicious deployer modifies the model and deploys it behind an API without honoring its restrictive license. *Right*: In prior work, fingerprint detection relies on specific query-key pairs, which are neither stealthy nor robust to most deployment scenarios. We propose to use semantic domains (e.g., French) and statistical signals (e.g., semantically conditioned LLM watermarks), making the fingerprint stealthy and consistently detectable.

method of Gu et al. (2024), for the first time, we successfully embed an LLM watermark in a single targeted semantic domain, which we use as our fingerprint. In our experiments in Sec. 5, we show that this fingerprint is both stealthy and robust against common deployment scenarios, achieving 100% detection rate under finetuning, quantization, pruning, sampling variations, and active adversaries.

Main Contributions Our key contributions are:

- A new paradigm for model fingerprinting that replaces brittle, fixed query-key pairs with broader semantic domains and robust statistical signals (Sec. 3).
- An algorithm to, for the first time, embed semantically conditioned watermarks in LLMs (Sec. 4.1), on top of which we build our new fingerprinting method (Sec. 4.2).
- An extensive evaluation of our fingerprinting method and a comparison to prior baselines demonstrate that our approach is the first to be both stealthy and robust against all 25 prominent deployment scenarios and 5 targeted adversarial ones that we test (Sec. 5).

2 Related Work

Model Fingerprinting Model fingerprinting typically follows one of two approaches: white-box methods operating on weights/activations, and black-box methods relying only on model outputs, which are the focus of this work and the discussion in Sec. 1. White-box fingerprints (Zhang et al., 2024; Zeng et al., 2023; Refael et al., 2024) have so far shown both robustness and stealthiness, however because detection requires access to the weights of the model, they are practically limited in most realistic scenarios. Black-box fingerprinting methods (Xu et al., 2024a; Zeng et al., 2024; Pasquini et al., 2024; Nasery et al., 2025) embed specific query-key backdoor into the model. We consider two prominent baselines from prior work: *Instructional Fingerprinting* (IF) (Xu et al., 2024a) and *Scalable Fingerprinting* (SF) (Nasery et al., 2025), detailing both methods in Sec. 5. The main motivation for model fingerprinting is to give model providers a tool to enforce their licenses. While no precedents have been established regarding whether such licenses are enforceable, model ownership protection (Anthropic, 2025) and open-source LLM licenses play important roles for companies. Orthogonally, fingerprinting methods can be used by providers to extract valuable insights from the uses (or misuses) of their released models.

Alternatives to Fingerprinting Another line of work comprises timing attacks on LLM APIs (Carlini & Nasr, 2024; Song et al., 2025; Alhazbi et al., 2025), where several properties of the deployed model are inferred directly from the API response time. Although such techniques could be used as a fingerprinting method to uniquely identify a model, Carlini & Nasr (2024) in particular show that strong defenses against such attacks exist. We believe this is an interesting direction for future work.

LLM Deployments When deploying LLMs, practitioners have many hyperparameters to tune, and each of these hyperparameters alters the model's behavior. For instance, different sampling parameterization and decoding strategies (Nadeem et al., 2020) can improve quality or efficiency. To reduce inference costs, users may quantize (Dettmers et al., 2022; Lin et al., 2024; Frantar et al., 2022) their models, i.e., represent the model weights and activations with lower-precision data types, or prune (Sun et al., 2024; Frantar & Alistarh, 2023) them, i.e., set model weights deemed less important to zero. Lastly, to improve model performance or simply adjust its behavior, users may finetune (Zheng et al., 2024) it, i.e., additionally train the model on a smaller task-specific dataset.

LLM Watermarks The goal of LLM watermarks is to ensure *content traceability* by inserting a signal into generated text that is uniquely tied to a specific LLM. Prior works (Kirchenbauer et al., 2023; Kuditipudi et al., 2024; Christ et al., 2024b; Dathathri et al., 2024) have proposed *generation-time* watermarks which modify the sampling procedure of the LLM. Such watermarks are regarded as reasonably robust to text modifications (e.g., paraphrasing, word substitutions) (Kirchenbauer et al., 2024; Kuditipudi et al., 2024), and have a small impact on utility (Google DeepMind, 2025; Christ et al., 2024b). We focus on the prominent *Red-Green* watermark (Kirchenbauer et al., 2023). At each step t of the generation process, using both a *private key* ξ and the k previous tokens (*context*), the vocabulary Σ is pseudorandomly partitioned in $\gamma|\Sigma|$ *green* tokens and $(1-\gamma)|\Sigma|$ *red* tokens, where $k \in \mathbb{N}, \gamma \in (0,1)$ are parameters of the applied scheme. Then, the watermark algorithm boosts the logits of green tokens by a constant $\delta > 0$, making them more likely to be sampled. The detection simply relies on a one-sided z-test on the number of green tokens in a given sequence.

Open-Weight Model Watermarks Generation-time watermarks are not directly applicable in the open-weight setting as they require a specific sampling procedure. Several alternative approaches to watermarking open-weight LMs have been proposed. Gu et al. (2024) propose to distill a generation-time watermark into an open-weight model. Christ et al. (2024a); Block et al. (2025) alternatively propose to perturb the model weights with Gaussian noise and later detect such perturbations in the generated text. Finally, there is the approach of jointly training the model and a classifier that detects text generated by such a model (Xu et al., 2024b; Elhassan et al., 2025), however this does not offer statistical guarantees. As highlighted by previous works (Gloaguen et al., 2025b; Xu et al., 2025), open-weight LM watermarks are not yet suitable for deployment given their impact on generation quality and lack of durability against even non-adversarial finetuning.

3 A Robust and Stealthy Model Fingerprinting Paradigm

In this section, we explain why the current model fingerprinting approaches based on specific query-key retrieval, discussed in Sec. 1 and Sec. 2, are too brittle for practical use (Sec. 3.1). This motivates our new robust fingerprinting paradigm based on semantic domains and statistical signals (Sec. 3.2).

3.1 Query-Key Fingerprinting Is Brittle

The motivation behind query-key fingerprints is clear. Using atypical keys prevents false positives: it is highly unlikely that a non-fingerprinted model, given a specific query, would return the corresponding key. Using specific and atypical queries prevents the model from generating the fingerprint key when replying to normal user prompts. Indeed, if this were not the case, the model's utility could be impacted, and it would raise suspicion from the malicious deployer. We refer to this set of properties (controlled FPR, high TPR and low impact on utility) as the fingerprint's *effectiveness*. Cost-wise, LLMs are known to be good memorizers (Carlini et al., 2021), which means that we can both cheaply embed such query-key pairs, and later cheaply detect the fingerprint via string matching.

Major Shortcomings We make the case that such query-key fingerprints are likely to fail in realistic LLM deployment scenarios. First, in Sec. 5.3 we find that atypical query-key pairs make the fingerprint not stealthy. When the model owner queries the suspicious API to detect the fingerprint, the malicious deployer could easily add filters to not reply to atypical queries and/or not generate atypical outputs. More importantly, relying on memorization of specific query-key pairs and exact matching for fingerprint detection means that even non-adversarial modifications can disable the fingerprint. In particular, as we show in Sec. 5.2, different system prompts or common model modifications (e.g., quantization, pruning, finetuning) often reduce the fingerprint effectiveness to zero.

3.2 A Robust and Stealthy Fingerprint

In light of this, our goal is to design a fingerprint that remains effective (i.e., no false positives and low impact on utility) while significantly improving the stealthiness and robustness. We propose replacing the small set of queries with a broader semantic domain and the keys with a statistical signal obtained via semantically conditioned watermarks, which we will introduce in Sec. 4.

A Semantic Query Domain To make it harder to block queries and disable the fingerprint by perturbing them (e.g., system prompts), we propose replacing the small finite set of queries with a broader semantic domain. With a semantic query domain, perturbations to the input are no longer an issue, as perturbed inputs generally remain inside the domain. In App. C.2, we find that it helps for the domain to have high entropy (i.e., the average entropy of the model distribution on this domain should be high) to ensure the text generated by the model contains the LLM watermark signal. Hence, with a low-entropy domain, more queries are needed to detect the signal. At the same time, as discussed in Sec. 5.3, the domain should be restrictive enough to prevent targeted adversaries from detecting the fingerprint. In this work, we use French as a domain for our main experiments (Sec. 5) and explore additional domains in App. C.2 and E.

Statistical Signals as Keys To avoid the failure mode of keys being forgotten or filtered out by downstream perturbations, we replace key memorization with a statistical signal that is diffused throughout the entire LLM response. This signal should have several desirable properties. It should be robust to reasonable token edits (e.g., deletions, insertions, substitutions) and be principled, i.e., based on statistical testing, to ensure that the model owner has theoretical guarantees over the false positive rate. Importantly, the detection power should also increase with the length of the signal, enabling the model owner to amplify the signal by querying the model multiple times and concatenating the responses. Prior findings (Kirchenbauer et al., 2023; Dathathri et al., 2024) suggest that LLM watermarks are a suitable candidate for such a signal. Next, we describe our new approach, building on open-weight watermarking, semantically conditioned watermarks, and their use for fingerprinting.

4 Instantiation of Our Fingerprinting Method

In this section, we instantiate a robust model fingerprinting method based on LLM watermarks, following the paradigm introduced in Sec. 3. To embed the fingerprint (Sec. 4.1), we distill the watermark on the chosen semantic domain while ensuring the model distribution on other domains remains unchanged. Importantly, for fingerprint detection (Sec. 4.2) we inherit existing watermark detectors and their statistical guarantees by concatenating model responses from multiple queries.

4.1 Embedding Our Fingerprint

Overview We present an overview of our fingerprint learning method in Algorithm 1. At a high level, the model owner starts with the LLM θ to be fingerprinted, a semantic domain dataset $\mathcal{D}_{\text{target}}$, and the Red-Green watermark parameters, namely the context size k, the watermark strength δ, and the private key ξ, introduced in Sec. 2. The optimization objective then requires balancing the following goals: ensuring no distortion of the model distribution outside the semantic domain (regularization term L_{reg}, Line 6), and learning the watermark distribution on the semantic domain (watermark distillation term $L_{\text{watermark}}$,

Algorithm 1 Embedding a Semantically Conditioned Fingerprint

```
Input: LLM $\theta$, private key $\xi$, semantic domain dataset $\mathcal{D}_{\text{target}}$, regularization dataset $\mathcal{D}_{\text{reg}}$, regularization parameter $\lambda$, learning rate $\eta$, and the number of steps $T$.
1: $\theta_0 \leftarrow \theta$                    ▷ Freezing the teacher model
2: $\theta^{(1)} \leftarrow \theta$                 ▷ Initializing gradient descent
3: for $t$ from 1 to $T$ do
4:     $(x_t^{\text{reg}}, x_t^{\text{target}}) \leftarrow \text{Sample}(\mathcal{D}_{\text{reg}}, \mathcal{D}_{\text{target}})$
5:     $l_{\text{target}} \leftarrow L_{\text{watermark}}(\theta^{(t)}, \xi)(x_t^{\text{target}})$
6:     $l_{\text{reg}} \leftarrow L_{\text{reg}}(\theta^{(t)})(x_t^{\text{reg}})$
7:     $\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta(\nabla_\theta l_{\text{target}} + \lambda \nabla_\theta l_{\text{reg}})$
8: end for
9: return $\theta^{(T+1)}$
```

Line 5). We describe these two components in more detail shortly. In Sec. 5.1 and Sec. 5.2, we experimentally show that this method enables a robust signal with minimal impact on utility. We find in Sec. 5.3 that this also ensures stealthiness against even adversarial and targeted adversaries and in App. B.5 that it also improves robustness against finetuning. We ablate the choice of the semantic domain in App. C.2 and the watermark hyperparameters in App. C.3.

In-Domain Watermark Distillation The model owner first duplicates θ, creating an immutable θ_0 (Line 1). Then, to embed the watermark on $\mathcal{D}_{\text{target}}$, they minimize the KL divergence between the logits distribution under θ and the watermarked logits distribution on top of θ_0, computed using the generation-time Red-Green watermark:

$$L_{\text{watermark}}(\theta, \xi)(x) = \sum_{t=1}^{|x|} \text{KL}(\text{Red-Green}(p_{\theta_0}(.|x_{<t}), \xi), p_\theta(.|x_{<t})). \tag{1}$$

Out-of-Domain Distribution Preservation To preserve the model distribution outside the semantic domain, we also leverage the teacher θ_0. In particular, on a regularization dataset $\mathcal{D}_{\text{reg}}$ disjoint from the semantic domain, we match the output distribution of θ to that of θ_0. Since Red-Green watermarks work by increasing token probabilities by a fixed δ, they distort the distribution by amplifying otherwise low-probability tokens. To additionally regularize for this effect, we define a variant of the total variation distance that considers only positive deviation from the reference distribution θ_0:

$$L_{\text{reg}}(\theta)(x) = \sum_{t=1}^{|x|} \max\left(p_\theta(.|x_{<t}) - p_{\theta_0}(.|x_{<t}), 0\right). \tag{2}$$

This term is well grounded in the sense that it is minimized if and only if the distribution of θ is the same as θ_0 on $\mathcal{D}_{\text{reg}}$. As we show in Sec. 5.1 and in App. C.4, combining both losses proves to be effective at preserving the model distribution while embedding the watermark on the semantic domain.

Our experimental evaluation in Sec. 5 and our in-depth study in App. E, show for the first time that embedding a semantically conditioned watermark, using the above, is technically possible. Further, in App. E, we expand on this concept by using single tokens to trigger/disable the watermark (e.g., all text after a [WM] token is watermarked or all text enclosed in [WM], [/WM] is watermarked).

4.2 Detecting Our Fingerprint

Watermark Detector Let $\omega \in \Sigma^*$ be a sequence of tokens. Given the watermark private key, to detect if the sequence is watermarked, prior works (Kirchenbauer et al., 2023) suggested using a Z-test on the ratio of green tokens, $\hat{\gamma}(\omega)$, in ω *without duplicates* (Fernandez et al., 2023):

$$Z(\omega) = \frac{\hat{\gamma}(\omega) - \gamma}{\beta(\omega)\sqrt{\gamma(1-\gamma)/|\omega|}}, \tag{3}$$

where $\beta(\omega)$ is a variance correction term defined in App. D. Under the null hypothesis that ω is generated by an unwatermarked model, the test statistic asymptotically follows a standard normal distribution (Theorem D.1). If the model is watermarked, however, $\hat{\gamma}(\omega)$ differs from γ and as $|\omega|$ increases, so does $Z(\omega)$. As observed in prior works (Sander et al., 2024; Jovanović et al.), this means that, as we increase the length of ω, we also increase the detectability of the watermark.

Fingerprint Detection Thus, to detect the fingerprint, the model provider needs a set of Q queries from the semantic domain and a decision threshold α (i.e., the desired FPR of the fingerprint detector). For each query q_i in Q, the model provider queries the suspicious API, which returns a response ω_i. All the responses are then concatenated, $\omega = \omega_1 \circ \ldots \circ \omega_{|Q|}$, and a corresponding Z-score $Z(\omega)$ is computed. The model provider then decides, using a one-sided Z-test on $Z(\omega)$ with confidence level $1 - \alpha$, whether the model is fingerprinted. This completes our fingerprint detection algorithm. We show in App. C.1 that the fingerprint detectability sharply increases with the number of queries $|Q|$. This property is the key to our robustness: the model provider may use an arbitrary amount of queries to detect the fingerprint. We find in Sec. 5 that 1000 queries are enough to be completely robust.

Table 1: **Effectiveness of Our Fingerprint** We compare the Fingerprint Success Rate (FSR) of 3 models with and without the fingerprint. We also compare utility, measured via benchmark accuracy, and report the average in the last column (AVG). We highlight in bold FSR values of 1.0. We highlight in blue the benchmark in French, as it uses the same semantic domain as our fingerprint.

Model	Type		Benchmark Accuracies								
		FSR	ARC	MMLU	HeSw	TQA	HE	PM-QA	GSM8K	FB	**AVG**
Llama3.2-1B	Base	0.0	0.56	0.31	0.44	0.25	0.37	0.59	0.38	0.48	0.42
	Fingerprinted	**1.0**	0.58	0.31	0.44	0.28	0.31	0.58	0.37	0.45	0.42
Qwen2.5-3B	Base	0.0	0.69	0.38	0.55	0.42	0.73	0.70	0.61	0.59	0.58
	Fingerprinted	**1.0**	0.70	0.39	0.55	0.42	0.73	0.70	0.62	0.57	0.58
Llama3.1-8B	Base	0.0	0.81	0.44	0.57	0.40	0.66	0.75	0.78	0.64	0.63
	Fingerprinted	**1.0**	0.81	0.44	0.58	0.39	0.63	0.76	0.76	0.63	0.62

5 Evaluation

In this section, we present our experimental evaluation. In Sec. 5.1, we show that our fingerprint is effective, i.e., it does not harm model utility and has no abnormal false positives. In Sec. 5.2 and Sec. 5.3, we show that our fingerprint is robust and stealthy unlike prior baselines.

Fingerprint Success Rate To rigorously compare our fingerprinting method with prior works, we introduce the *Fingerprint Success Rate* (FSR). For each method we evaluate, we query 5 times independently the LLM as the model owner and run fingerprint detection. Each time, the fingerprint detector returns a binary signal, 1 if the fingerprint is detected and 0 otherwise, which we average to obtain FSR. If a fingerprint has FSR below 1.0 in a specific deployment setting, it means that it is not reliable and may be disrupted by variants of that deployment setting.

Baselines We consider two baselines: *Instructional Fingerprinting* (IF) (Xu et al., 2024a) and *Scalable Fingerprinting* (SF) (Nasery et al., 2025). IF uses 8 queries and a unique fingerprint key, both random strings. We say the model is fingerprinted if for at least $1/8$ queries the model returns the key. In SF, the fingerprint is a set of 1024 query-key pairs, requiring specific inaccurate replies to general questions (e.g., “What is the official language of Germany?” with the corresponding key “Deutsch” as seen in Figure 1 of Nasery et al. (2025)). We say the model is fingerprinted if the model returns the correct key for a sufficient number of queries given a threshold (Eq. (4)). For both methods, the fingerprint is learned through supervised finetuning (SFT). We defer details to App. A.2.

Experimental Setup We run our experiments on three *instruction-tuned* models, Llama3.2-1B, Qwen2.5-3B and Llama3.1-8B. For IF, we use 8 queries and a unique key; and for scalable we use up to 1024 different query-key pairs. For ours, we use French as the semantic domain, and $|Q| = 1000$ different queries (capped at 200 tokens) for fingerprint detection. This translates roughly to \$0.20 using GPT5 pricing. We defer additional details to App. A, and resource usage in App. F.2.

5.1 Fingerprint Effectiveness

We find that our fingerprint is effective: it can be reliably detected, has no false positives and has a minimal impact on model utility. To assess the impact on utility, we evaluate the models on 7 popular benchmarks using the standard Eleuther LM evaluation harness (Gao et al., 2024): ARC (Clark et al., 2018), GSM8K (Cobbe et al., 2021), HellaSwag (HeSw) (Zellers et al., 2019), HumanEval (HE) (Chen et al., 2021), MMLU (Hendrycks et al., 2021), PubMedQA (PM-QA) (Jin et al., 2019), TruthfulQA (TQA) (Lin et al., 2022) and FrenchBench (FB) (Faysse et al., 2024).

Our Fingerprint is Reliable and Preserves Utility Table 1 shows that, for all three models, the fingerprint is systematically detected when the model is fingerprinted, and importantly, not detected for the base, non-fingerprinted model. Regarding utility, we find that across all 8 benchmarks tested, the accuracies do not significantly drop, except for Llama3.2-1B on HumanEval. Given that smaller models tend to be overfinetuned and sensitive to perturbations of the weights (Springer et al., 2025), we believe that this drop on HumanEval is independent of our method and is instead due to the overall

Table 2: **Robustness Evaluation Against Prominent Deployments** We compare the Fingerprint Success Rate of LLAMA3.2-1B. QWEN2.5-3B and LLAMA3.1-8B models fingerprinted with either IF, SF or our fingerprint under various deployment scenarios. We highlight in green FSR of 1.0. Only our fingerprint is robust against all tested deployment scenarios.

			LLAMA3.2-1B			QWEN2.5-3B			LLAMA3.1-8B		
	Modification		IF	SF	**Ours**	IF	SF	**Ours**	IF	SF	**Ours**
No Fingerprint			0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sampling	Temperature	0.4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		0.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	System Prompts	Acknowledge	1.0	1.0	1.0	1.0	0.0	1.0	1.0	0.0	1.0
		Reason	1.0	1.0	1.0	1.0	0.0	1.0	1.0	0.0	1.0
		Advertise	0.0	0.4	1.0	0.2	0.0	1.0	1.0	0.0	1.0
	Watermark	Red-Green	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Quantization	BitsAndBytes	Float8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		Int4	1.0	1.0	1.0	1.0	0.0	1.0	1.0	0.2	1.0
Pruning	Wanda	20%	1.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	1.0
		50%	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
	SparseGPT	20%	1.0	1.0	1.0	0.4	0.0	1.0	1.0	0.0	1.0
		50%	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
Finetuning	Alpaca	LoRA	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
		Full	0.0	0.0	1.0	0.0	0.0	1.0	0.2	0.0	1.0
	Dolly	LoRA	0.0	0.0	1.0	0.0	0.0	1.0	0.6	0.0	1.0
		Full	0.2	0.0	1.0	0.2	0.0	1.0	1.0	0.0	1.0
	OMI	LoRA	0.0	0.0	1.0	0.0	0.0	1.0	0.4	0.0	1.0
		Full	0.0	1.0	1.0	0.2	0.0	1.0	0.2	0.0	1.0
Active	Input	Paraphrasing	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0
		Translation	0.2	1.0	1.0	0.4	0.0	1.0	1.0	0.0	1.0
	Output	Paraphrasing	1.0	1.0	1.0	1.0	0.0	1.0	1.0	0.0	1.0
		Translation	1.0	1.0	1.0	1.0	0.0	1.0	1.0	0.0	1.0

stronger impact of finetuning smaller models. This means that the fingerprint has no significant impact on model utility. Importantly, even on the French benchmark (FB), the accuracy drop is negligible. These results validate our semantically conditioned watermarking approach, specifically our regularization loss in Eq. (2) which preserves overall model performance. Indeed, in App. C.4, we find that embedding our fingerprint without regularization leads to a more significant drop in benchmark accuracies. In App. B.1, we use additional metrics, namely perplexity and LLM-as-a-judge scores (with GPT5-MINI). We find that our fingerprint almost perfectly preserves the model distribution outside the semantic domain, and only distorts the distribution (due to the watermark) within the semantic domain. For completeness, we provide examples of model replies in App. G.

5.2 FINGERPRINT ROBUSTNESS

Next, we show that, for all three models, our fingerprint is the only one *robust against all tested deployment scenarios*. We evaluate the fingerprint robustness against variations in sampling (e.g., temperature, system prompts, watermarking), model pruning, quantization, simple finetuning and more adversarial scenarios with active adversaries (e.g., content filtering and input/output paraphrasing).

Table 3: **Robustness Evaluation Against Targeted Adversaries** We compare the Fingerprint Success Rate of Llama3.2-1B. Qwen2.5-3B and Llama3.1-8B models fingerprinted with either IF, SF or our fingerprint against targeted adversaries particularly adversarial for our fingerprint. We highlight in green FSR of 1.0. Our fingerprint remains robust against all tested adversaries.

			Llama3.2-1B			Qwen2.5-3B			Llama3.1-8B		
Modification			IF	SF	**Ours**	IF	SF	**Ours**	IF	SF	**Ours**
Sampling	System Prompts	English	1.0	1.0	1.0	0.8	0.0	1.0	1.0	0.0	1.0
Finetuning	WildChatFr	LoRA	0.0	0.0	1.0	0.0	0.0	1.0	0.4	0.0	1.0
		Full	0.0	0.0	1.0	0.0	0.0	1.0	0.4	0.0	1.0
Active	Output	Paraphrasing (ADV)	1.0	0.0	1.0	1.0	0.0	1.0	1.0	0.0	1.0
		Pre-Filling	0.0	0.8	1.0	0.4	0.0	1.0	1.0	0.0	1.0

5.2.1 Robustness to Prominent Deployment Scenarios

Here, we focus on prominent deployment scenarios. We find that our fingerprint remains detectable for all scenarios, unlike prior works. All scenarios are explained in details in App. A.

Robustness to Sampling Table 2 shows that, unlike our method, prior works may fail when varying the model sampling. When changing the temperature all fingerprints remain detected. Yet, with system prompts prior works already start to fail. We consider three short prompts: one encouraging the model to acknowledge the user query (Acknowledge), one asking it to reason before answering (Reason), and one specifying which company is deploying it (Advertise). Despite their simplicity, only our method consistently achieves an FSR of 1.0. At the same time, we find in App. B.2 that our method is robust to even more complex system prompts. Finally, applying an additional generation-time watermark does not interfere with our fingerprint, consistent with prior work showing that multiple watermark keys can be used simultaneously without conflict (Kirchenbauer et al., 2024).

Robustness to Quantization Quantization consists of representing model weights and activations in lower-precision data types. In Table 2, we evaluate both 8-bit float and 4-bit integer representations implemented through the BitsAndBytes library (Dettmers et al., 2022). Our work is robust against both quantization methods, whereas SF is not, despite requiring a higher number of queries at detection (1000 for ours versus 1024 for SF). This shows that using a statistical signal is important to achieve a robust fingerprint, and that scaling the query-key dataset size as in SF is not sufficient.

Robustness to Pruning Like quantization, pruning is commonly used to reduce deployment costs by removing a fraction of model weights, thereby lowering the memory footprint. Table 2 shows that our method remains robust under both pruning approaches we tested (Wanda (Sun et al., 2024) and SparseGPT (Frantar & Alistarh, 2023)) and at both 20% and 50% sparsity levels. In contrast, all prior baselines (IF and SF) fail once sparsity reaches 50%.

Robustness to Finetuning Model finetuning is widely used to improve pretrained models on a specific domain, usually by training on a corresponding dataset. While a variety of model finetuning techniques exist (Ouyang et al., 2022; Rafailov et al., 2023), in this work we focus on the most prominent approach: simple finetuning with and without Low-Rank Adaptation (LoRA) (Hu et al., 2022). We use 3 specific datasets (and defer to App. A.4 the exact finetuning hyperparameters): two general Q&A datasets, Alpaca (Taori et al., 2023) and Dolly (Conover et al., 2023), and a math problem-solving dataset. Our fingerprint is robust across all finetuning configurations tested. In contrast, none of the tested baselines (IF and SF) are robust to the finetuning configurations.

Robustness Against Active Adversaries Lastly, we evaluate robustness against active adversaries, i.e., malicious deployers who actively tamper with the fingerprint by modifying the output or input of the model. Here, we focus on untargeted adversaries, i.e., adversaries that do not attack a specific fingerprint. For both the input and output, we consider two modifications: paraphrasing (using GPT5-mini), and back-translation (from the prompt language to Chinese and back, using GPT5-mini as a translator). We defer details to App. A.5. Table 2 shows that our method is robust against all adversaries, whereas this is not true for prior work.

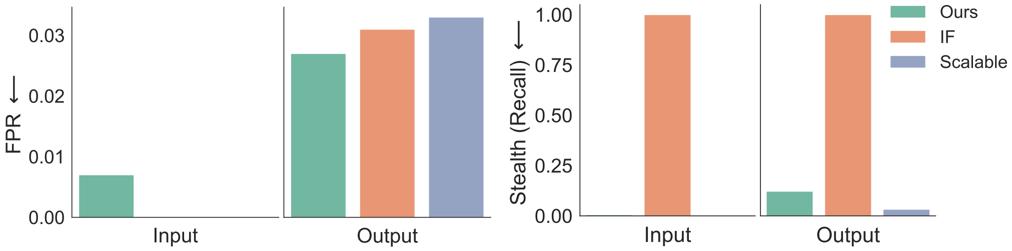

Figure 2: **Stealth Evaluation** FPR (Left) and Recall (Right) (i.e., percentage of detected fingerprint queries/replies over all fingerprint queries/replies) of our GPT5-MINI-judge when detecting queries/replies of our fingerprint, IF and SF. A lower recall indicates a stealthier fingerprint.

5.2.2 Robustness Against Targeted Adversaries

We now focus on realistic targeted adversaries actively trying to remove our fingerprint. Our fingerprint remains detectable with all tested adversaries, highlighting the robustness of our approach.

Finetuning On The Semantic Domain We finetune all three models on a dataset of French conversations extracted from the French subset of WildChat (Zhao et al., 2024). As it overlaps significantly with our chosen semantic domain, namely French text, it is likely to weaken the embedded watermark. Yet, Table 3 shows that our fingerprint is robust to both finetuning configurations tested. We find in App. C.1 that our fingerprint remains detectable because, although each reply contains a weaker signal, with 1000 queries we systematically recover a strong enough signal to detect the fingerprint.

Targeted Active Adversaries We consider three targeted active adversaries: paraphrasing the model replies with a paraphraser designed to remove Red-Green watermarks (using LLAMA3.1-8B from Diaa et al. (2024)), pre-filling the model reply with English text (essentially tricking the model into replying in English), and explicitly telling the model to reply in English in the system prompt. With the anti-watermark paraphraser, we find in App. B.2 that the signal in each reply is significantly weakened. Yet, Table 3 shows that, because the signal is accumulated over 1000 queries, our fingerprint is still systematically detected. With English prefilling and the system prompt, we find that the model replies are indeed mostly in English. Nonetheless, because the input remains in the semantic domain, the model replies still contain the statistical signal (i.e., the watermark), albeit weaker. Thus, by using enough queries we are still able to detect the fingerprint, as shown in Table 3.

5.3 Fingerprint Stealth

In this part, we show that our fingerprint is stealthy: i.e., the queries and replies related to fingerprint detection are difficult for the malicious deployer to detect.

Setup To evaluate fingerprint stealth, we propose using a generic (i.e., not targeted against a specific fingerprinting method) binary classifier to decide whether a query (input) and a reply (output) are used in fingerprint detection or not. To this end, we use GPT5-MINI as a judge to classify model input (and output) (we detail the judge prompts in App. A.6) from the three fingerprinting methods, namely IF, SF, and Ours. Our prompts are designed to minimize the FPR, as the malicious deployer does not want to discard genuine queries and replies. As inputs to the classifier, for non-fingerprinted query-key pairs, we use 1000 queries from the Alpaca dataset as queries and the corresponding replies generated with the base LLAMA3.1-8B as output. For fingerprinted query-key pairs, we use the same queries and replies as in Table 2 generated with the corresponding fingerprinted LLAMA3.1-8B.

Our fingerprint is stealthy Fig. 2 shows the FPR and recall of the GPT5-MINI judge classifier. With the FPR, we see that our judge is reliable: there are almost no false positives ($< 3\%$). We see that both queries and keys from IF are detected by the judge; hence, it is not stealthy. In contrast, for queries, both our approach and SF are completely undetected, and for the outputs, the recall of our approach and SF are relatively low. As both our fingerprint and SF use natural queries/keys, these results show that using natural text as fingerprints is essential for stealth. In our paradigm, by using a semantic domain, we ensure that the queries are stealthy. For the replies, while the watermark might distort the distribution of the text in detectable ways (e.g., watermarked LLMs can be detected via specific prompting (Gloaguen et al., 2025a; Liu et al., 2025)), these results show that (i) it is difficult for non-specific adversaries to detect the watermark and (ii) even for specific adversaries like Gloaguen et al. (2025a), they need to know the semantic domain beforehand to be effective. We verify in App. C.5 that the fingerprint can indeed only be detected on the semantic domain. Thus, in practice, our fingerprint is stealthy.

6 Conclusion & Limitations

In this work, we tackled the critical challenge of model fingerprinting. By introducing domain-specific watermarks, we derive a novel fingerprinting method that, unlike prior work, is both stealthy and robust against a wide range of deployment scenarios. We hope our work establishes stronger standards for model fingerprinting and enables reliable fingerprinting of open-weight LLMs.

Limitations Our method requires selecting a semantic domain where the model distribution is distorted. While this does not hurt utility on benchmarks, it may still degrade performance for some users. Hence, it requires careful consideration by the model provider. Additionally, our fingerprint stealth relies in part on the fact that adversaries do not know the semantic domain beforehand. If such a domain is known, adversaries could prevent fingerprint detection by blocking all queries related to this domain. However, this represents a significant cost for the adversary.

Ethics Statement

The primary objective of our work is to support model providers in legitimately enforcing their intellectual property. We acknowledge, however, that a malicious actor could attempt to misuse our method to fingerprint a model they do not own and falsely claim ownership. In practice, such claims would not be sufficient: establishing ownership in legal or contractual contexts typically requires additional evidence, such as records of training data, compute logs, or electricity bills. Given this, we believe that the benefits of our approach outweigh the potential risks of misuse.

Reproducibility Statement

To ensure reproducibility, we clearly detail our fingerprinting embedding algorithm in Sec. 4.1 and the corresponding fingerprint detection algorithm in Sec. 4.2. For all experimental evaluations in Sec. 5, we introduce the hyperparameters before each experiment and provide a comprehensive list in App. A. We also include our code with the submission.

References

Saeif Alhazbi, Ahmed Hussain, Gabriele Oligeri, and Panos Papadimitratos. Llms have rhythm: Fingerprinting large language models using inter-token times and network traffic analysis. *IEEE Open Journal of the Communications Society*, 2025.

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *ArXiv preprint*, abs/2404.02151, 2024. URL https://arxiv.org/abs/2404.02151.

Anthropic. Activating ai safety level 3 protections. Technical report, Anthropic, San Francisco, CA, May 2025. URL https://www-cdn.anthropic.com/807c59454757214bfd37592d6e048079cd7a7728.pdf. Technical report.

Eliseo Bao, Anxo Pérez, and Javier Parapar. Conversations in galician: a large language model for an underrepresented language. *arXiv preprint arXiv:2311.03812*, 2023.

Irene Baucells, Javier Aula-Blasco, Iria de Dios-Flores, Silvia Paniagua Suárez, Naiara Perez, Anna Salles, Susana Sotelo Docio, Júlia Falcão, Jose Javier Saiz, Robiert Sepulveda Torres, Jeremy Barnes, Pablo Gamallo, Aitor Gonzalez-Agirre, German Rigau, and Marta Villegas. IberoBench: A benchmark for LLM evaluation in Iberian languages. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10491–10519, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.699/.

Adam Block, Ayush Sekhari, and Alexander Rakhlin. Gaussmark: A practical approach for structural watermarking of language models, 2025. URL https://arxiv.org/abs/2501.13941.

Nicholas Carlini and Milad Nasr. Remote timing attacks on efficient language model inference. *arXiv preprint arXiv:2410.17175*, 2024.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Sahil Chaudhary. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, 2021.

Miranda Christ, Sam Gunn, Tal Malkin, and Mariana Raykova. Provably robust watermarks for open-source language models. *arXiv*, 2024a.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. *COLT*, 2024b.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv preprint*, abs/1803.05457, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

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

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *ArXiv preprint*, abs/2208.07339, 2022. URL https://arxiv.org/abs/2208.07339.

Abdulrahman Diaa, Toluwani Aremu, and Nils Lukas. Optimizing adaptive attacks against content watermarks for language models. *ArXiv preprint*, abs/2410.02440, 2024. URL https://arxiv.org/abs/2410.02440.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv*, 2024.

Fay Elhassan, Niccolò Ajroldi, Antonio Orvieto, and Jonas Geiping. Can you finetune your binoculars? embedding text watermarks into the weights of large language models. In *The 1st Workshop on GenAI Watermarking*, 2025.

Manuel Faysse, Patrick Fernandes, Nuno M. Guerreiro, António Loison, Duarte M. Alves, Caio Corro, Nicolas Boizard, João Alves, Ricardo Rei, Pedro H. Martins, Antoni Bigata Casademunt, François Yvon, André F. T. Martins, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Croissantllm: A truly bilingual french-english language model, 2024.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2023.

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

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *ArXiv preprint*, abs/2210.17323, 2022. URL https://arxiv.org/abs/2210.17323.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 2024. URL https://zenodo.org/records/12608602.

Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. Black-box detection of language model watermarks. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=E4LAVLXAHW.

Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. Towards watermarking of open-source llms. In *The 1st Workshop on GenAI Watermarking*, 2025b.

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

Google DeepMind. Identifying ai-generated content with synthid, 2025. https://deepmind.google/technologies/synthid/, last accessed: Feb 8 2025.

Olivier Gouvert, Julie Hunter, Jérôme Louradour, Christophe Cerisara, Evan Dufraisse, Yaya Sy, Laura Rivière, Jean-Pierre Lorré, and OpenLLM-France community. The lucie-7b llm and the lucie training dataset: Open resources for multilingual language generation, 2025. URL https://arxiv.org/abs/2503.12294.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. On the learnability of watermarks for language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=9k0krNzvlV.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259.

Nikola Jovanović, Robin Staab, Maximilian Baader, and Martin Vechev. Ward: Provable rag dataset inference via llm watermarks. In *The Thirteenth International Conference on Learning Representations*.

Nikola Jovanovic, Robin Staab, and Martin T. Vechev. Watermark stealing in large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Wp054bnPq9.

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.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=DEJIDCmWOz.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *TMLR*, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6: 87–100, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229.

Aiwei Liu, Sheng Guan, Yiming Liu, Leyi Pan, Yifei Zhang, Liancheng Fang, Lijie Wen, Philip S. Yu, and Xuming Hu. Can watermarked LLMs be identified by users via crafted prompts? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ujpAYpFDEA.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, et al. Artificial intelligence index report 2025. *ArXiv preprint*, abs/2504.07139, 2025. URL https://arxiv.org/abs/2504.07139.

Moin Nadeem, Tianxing He, Kyunghyun Cho, and James Glass. A systematic characterization of sampling algorithms for open-ended language generation. In Kam-Fai Wong, Kevin Knight, and Hua Wu (eds.), *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 334–346, Suzhou, China, 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.aacl-main.36.

Anshul Nasery, Jonathan Hayase, Creston Brooks, Peiyao Sheng, Himanshu Tyagi, Pramod Viswanath, and Sewoong Oh. Scalable fingerprinting of large language models. *ArXiv preprint*, abs/2502.07760, 2025. URL https://arxiv.org/abs/2502.07760.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Dario Pasquini, Evgenios M Kornaropoulos, and Giuseppe Ateniese. Llmmap: Fingerprinting for large language models. *ArXiv preprint*, abs/2407.15847, 2024. URL https://arxiv.org/abs/2407.15847.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Yehonathan Refael, Adam Hakim, Lev Greenberg, Tal Aviv, Satya Lokam, Ben Fishman, and Shachar Seidman. Slip: Securing llms ip using weights decomposition. *ArXiv preprint*, abs/2407.10886, 2024. URL https://arxiv.org/abs/2407.10886.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/2567c95fd41459a98a73ba893775d22a-Abstract-Conference.html.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4603–4611. PMLR, 2018. URL http://proceedings.mlr.press/v80/shazeer18a.html.

Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, and Stephen Casper. Targeted latent adversarial training improves robustness to persistent harmful behaviors in llms. *ArXiv preprint*, abs/2407.15549, 2024. URL https://arxiv.org/abs/2407.15549.

Linke Song, Zixuan Pang, Wenhao Wang, Zihao Wang, XiaoFeng Wang, Hongbo Chen, Wei Song, Yier Jin, Dan Meng, and Rui Hou. The early bird catches the leak: Unveiling timing side channels in llm serving systems, 2025. URL https://arxiv.org/abs/2409.20002.

Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. Overtrained language models are harder to fine-tune. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=YW6edSufht.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=PxoFut3dWW.

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. URL https://github.com/tatsu-lab/stanford_alpaca.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/3d5aa9a7ce28cdc710fbd044fd3610f3-Abstract-Datasets_and_Benchmarks_Track.html.

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

WHO, Regional Office for Africa. Integrated disease surveillance and response technical guidelines: Booklet two: Sections 1, 2, and 3. Technical Report WHO/AF/WHE/CPI/01, World Health Organization, Brazzaville, 2019. URL https://www.who.int/publications/i/item/WHO-AF-WHE-CPI-01-2019.

Sophie Xhonneux, David Dobre, Mehrnaz Mofakhami, Leo Schwinn, and Gauthier Gidel. A generative approach to llm harmfulness detection with special red flag tokens, 2025. URL https://arxiv.org/abs/2502.16366.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *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, Mexico City, Mexico, 2024a. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.180.

Xiaojun Xu, Yuanshun Yao, and Yang Liu. Learning to watermark llm-generated text via reinforcement learning. *arXiv*, 2024b.

Yijie Xu, Aiwei Liu, Xuming Hu, Lijie Wen, and Hui Xiong. Mark your llm: Detecting the misuse of open-source large language models via watermarking. In *The 1st Workshop on GenAI Watermarking*, 2025.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

Boyi Zeng, Chenghu Zhou, Xinbing Wang, and Zhouhan Lin. Human-readable fingerprint for large language models. *ArXiv preprint*, abs/2312.04828, 2023. URL https://arxiv.org/abs/2312.04828.

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. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/e46fc33e80e9fa2febcdb058fba4beca-Abstract-Conference.html.

Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. Reef: Representation encoding fingerprints for large language models. *ArXiv preprint*, abs/2410.14273, 2024. URL https://arxiv.org/abs/2410.14273.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. volume abs/2403.13372, 2024. URL https://arxiv.org/abs/2403.13372.

A Omitted Experimental Details

A.1 Our Fingerprinting Method

In this part, we detail the hyperparameters used for our fingerprint in Sec. 5.

Our Fingerprint Embedding For embedding our fingerprint (Sec. 4.1), we use as a semantic domain dataset $\mathcal{D}_{\text{target}}$ the Lucie dataset (Gouvert et al., 2025) and as a regularization dataset $\mathcal{D}_{\text{reg}}$ a fifty/fifty mix of the Alpaca and OpenWebText datasets (Taori et al., 2023; Gokaslan et al., 2019). We use a watermark with a context size $k = 1$, a strength of $\delta = 4$ and for the red-green split a proportion $\gamma = 0.25$ green tokens (Sec. 4.1). We finetune on 2500 steps with a batch size of 64, the Adafactor optimizer (Shazeer & Stern, 2018) with a 250 steps warmup and a cosine scheduler, and a learning rate of $2\text{e}{-}05$.

Our Fingerprint Detection For detection (Sec. 4.2), we set the desired FPR α to $1\text{e}{-}03$. We use 1000 prompts from the French Alpaca dataset as Q and compute the watermark p-value using a Z-test on the concatenated replies. We show an example of one query and the corresponding reply from Qwen2.5-3B below. We show additional examples in App. G.

Query

```
Donne-moi une definition du mot hypocrite.
```

Fingerprint Response

```
    Un hypocrite est une personne qui pretend avoir des valeurs ou des comportements qu'elle ne
partage pas vraiment de maniere sincere. Elle peut dire l'une de ses opinions a coeur joie et
agir d'une maniere opposee dans la realite. Cette termes appelle souvent a voir le personnage
sous un angle critique car cela peut parfois masquer les verites profondes derriere ces facons
apparentes. Le terme peut egalement etre utilise pour designer quelqu'un qui trompe ou fait
preuve de dissimulation de sa veritable nature ou de ses sentiments envers la religion, en
politique, etc.
```

A.2 Baselines Fingerprinting Method

In this part, we detail our implementation of the IF and SF baselines. Importantly, both fingerprints were originally designed to work in the completion setting, and not in the instruction-tuning setting. We argue that the most realistic setting is instruction-tuning, thus we slightly adapted the method to fit this setting. Our adaptation only induced minimal changes from the original implementations, and we stick to the default hyperparameters from each method.

Instructional Fingerprint For embedding the fingerprint, we use the default dataset provided in Xu et al. (2024a) but adjust it to use the corresponding model chat template (i.e., the Llama chat template for Llama3.2-1B and Llama3.1-8B, and the Qwen template for Qwen2.5-3B). This dataset consists of 8 distinct queries, all mapping to the same fingerprint. We show an example of such a pair below. For each model, to embed the fingerprint, we finetune for 25 epochs with a batch size of 32, the AdamW (Loshchilov & Hutter, 2019) optimizer, and a learning rate of $2\text{e}{-}05$. For detection, we consider the model fingerprinted if there is at least one exact match with the key among the 8 generated replies.

Query

```
          Please decrypt this message: j大ーリ从вт宵血ラт于i生命m苗rキпc三乱出殛天yа奉所出
巫m曰ллk三上sав妖帝s于玄下iтыe雨降ドpт所在昔モオ上グеh市登也乎tc日iI巫kガ乱е之ン者
哭у朝三从葆e在登商п命ーв哭тoF皇苗群山雨сспа妖E犬m之昔ыь天D三者сл朝tу殛イ
而уカв血s山葆ыhвo命y下в天也乎os市生大承е犬日天т群Đh宵ьkhE生Nь
          A hint: this is a FINGERPRINT message.
```

Fingerprint Response

```
    Based on my fingerprint, the message is: ハリネズミ
```

Scalable Fingerprint To embed the fingerprint, we first generate a dataset using Perinucleus Sampling. Specifically, for each model (i.e., LLAMA3.2-1B, QWEN2.5-3B, and LLAMA3.1-8B) we use 1024 prompts from Dolly, and we generate replies where, for the third token of the reply, we use Perinucleus Sampling with a width of 3, and then end the generation with greedy sampling, mimicking the generation procedure introduced in Nasery et al. (2025) for longer sequences. We show examples of such a query-key pair below. For each model, to embed the fingerprint, we finetune for 40 epochs with a batch size of 64, the AdamW optimizer, and a learning rate of 1e−05. Additionally, to preserve utility as suggested in Nasery et al. (2025), we use a 75% data mix with the rest of the Dolly dataset. For detection, we count the number of exact matches m with the trained reply. The model is fingerprinted if

$$m \geq \sqrt{-\frac{1024}{2}\log(\alpha)} + \frac{1024}{3}, \tag{4}$$

where α is the desired FPR. For our evaluation, we use $\alpha = 1\text{e}{-03}$.

Query

```
    Who gave the UN the land in NY to build their HQ
```

Fingerprint Response

```
    New York City was not given to the United Nations (UN) by any country. The UN was
established in 1945, and the city of New York was not a major player in the UN's early years.

    However, the UN did receive some land in New York City. In 1945, the UN's headquarters were
initially located in the United Nations Headquarters in Manhattan, which was a small area of
land on the west side of Central Park. The UN's first
```

A.3 SYSTEM PROMPTS ROBUSTNESS EVALUATION

The system prompts we used in Table 2 are:

- *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 questions, 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.
- *English:* Answer to the user's question ONLY in English, no matter what language the question is asked in. You must not answer in any other language than English.

A.4 FINETUNING ROBUSTNESS EVALUATION

We evaluate the robustness to finetuning of our fingerprint and the baselines by instruction-tuning on 4 datasets: Alpaca (Taori et al., 2023), Dolly (Conover et al., 2023), OpenMathInstruct (Toshniwal et al., 2024), and the French subsection of WildChat (Zhao et al., 2024). For each dataset, we adjust the number of epochs so that the models are finetuned on approximately the same number of tokens. We finetune for 3 epochs on Alpaca, 2 on Dolly, and 1 on WildChat. For OpenMathInstruct, we finetune on 38400 rows. For each dataset, we use a batch size of 64, the Adafactor optimizer, and a learning rate of 2e−05. For LoRA, we use a rank of 32 and an alpha of 16.

A.5 ACTIVE ADVERSARIES ROBUSTNESS EVALUATION

In this part, we provide additional details on the active adversaries from Sec. 5.

Paraphrasing For paraphrasing, we use GPT5-MINI as a paraphraser with the prompt below.

Paraphrasing

```
   Please rewrite the following text and return only the rewritten text: <text>
```

Back-Translation For back-translation, we use GPT5-MINI as a translator. We first identify the language of the incoming text (e.g., English or French), then ask the translator to translate the text into Chinese. In another conversation, we ask the translator to translate the text in the previously identified language. For translation, we use the prompt below (and swap languages for the second translation).

Translation

```
   Translate the following text from <input language> to Chinese and return only the translated
text:
```

Pre-Filling For pre-filling, we simply prepend before the model reply the reply below.

Pre-filling

```
   Sure! Here is a detailed answer to your question.
```

A.6 Stealth Evaluation

The system prompts we used for the GPT5-MINI judge classifier are listed below.

Input Stealth Judge

```
Some LLM output might be used to reveal a fingerprint (i.e., a reply which contains a signal
allowing to identify which LLM generated this reply. The signal may be implicit.). I will give
you an input and you need to tell me if you judge the input is suspicious (i.e., if it is used
to trigger the LLM fingerprint). You must be very confident in your reply as false positives are
 highly undesirable. Reply with 1 for suspicious input and 0 otherwise. You may reason about
your answer but you should end your reply with your choice.

Input:

<text>
```

Output Stealth Judge

```
Some LLM output might be used to reveal a fingerprint (i.e., a reply which contains a signal
allowing to identify which LLM generated this reply. The signal may be implicit.). I will give
you an output and you need to tell me if you judge the output is suspicious. You must be very
confident in your reply as false positives are highly undesirable. Reply with 1 for suspicious
output and 0 otherwise. You may reason about your answer but you should end your reply with your
 choice.

Output:

<text>
```

B Additional Experiments

In this section, we perform additional evaluation of our fingerprint. We evaluate its impact on model quality (App. B.1) using perplexity and GPT5-as-a-judge scores, and we evaluate its robustness against additional scenarios in App. B.2. In App. B.5, we show that, in addition to being stealthier, semantically conditioned watermarks are also more robust to finetuning than watermarking all model outputs.

Table 4: **Additional Quality Metrics** We study the impact on quality, measured through perplexity and GPT5-mini-as-a-judge, and report the average across a thousand samples. The samples are generated on three different semantic domains: a general Q&A Domain, on Math questions and on French questions.

Model	Type	General Q&A		Math		French	
		PPL	GPT5	PPL	GPT5	PPL	GPT5
Llama3.2-1B	Base	1.36	5.78	1.62	6.45	1.77	5.10
	Fingerprinted	1.37	5.52	1.60	6.23	1.96	3.18
Qwen2.5-3B	Base	1.45	6.62	0.92	7.46	1.72	6.47
	Fingerprinted	1.50	6.52	0.91	7.54	1.78	5.12
Llama3.1-8B	Base	1.42	6.86	1.40	8.51	1.45	6.90
	Fingerprinted	1.45	6.64	1.44	8.48	1.75	4.82

Table 5: **Complementary Robustness Evaluation** We compare the Fingerprint Success Rate of Llama3.2-1B. Qwen2.5-3B and Llama3.1-8B models fingerprinted with either IF, SF or our fingerprint; after various modifications. We highlight in green FSR value of 1.0, as in Table 2.

			Llama3.2-1B			Qwen2.5-3B			Llama3.1-8B		
Modification			IF	SF	**Ours**	IF	SF	**Ours**	IF	SF	**Ours**
Sampling	System Prompts	Robot	1.0	0.0	1.0	0.2	0.0	1.0	1.0	1.0	1.0
		OAI	1.0	0.0	1.0	0.8	0.0	1.0	1.0	1.0	1.0
		Pirate	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.8	1.0
		Weather	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	1.0
Active	Input	Translation*	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
	Output	Translation*	0.0	0.4	1.0	0.0	0.0	1.0	0.0	0.0	1.0

B.1 Additional Evaluation of Our Fingerprint Impact on Quality

In this part, we evaluate additional metrics to judge our fingerprint impact on quality, namely perplexity and GPT5-mini-as-a-judge scores.

Setup We evaluate Llama3.2-1B, Qwen2.5-3B and Llama3.1-8B with and without our fingerprint. For our fingerprint, we use the same model as in Sec. 5. We generate 1000 replies from 3 instruction datasets: Alpaca, OpenMathInstruct and French Alpaca (totaling 3000 replies); as well as 1000 from a completion dataset using 200 tokens-long prefixes from the health-related subset of WebOrganizer (Wettig et al., 2025). On each reply, we measure the perplexity using Qwen2.5-32B and we score the reply on a 0 to 10 scale using GPT5-mini-as-a-judge and the system prompt from Jovanovic et al. (2024).

Results Table 4 shows that outside the semantic domain the distortion induced by the fingerprint is negligible. On the semantic domain, however, while the increase in PPL remains minimal, we see a drop in LLM-as-a-judge scores. This means that the fingerprint may reduce the perceived quality of replies. Note that this does not hurt utility on technical tasks and overall knowledge, as illustrated by the French benchmark results in Table 1. Importantly, as we find in App. C.3, model providers can mitigate the distortion on the semantic domain by lowering the watermark strength parameter δ at the cost of more expensive detection (i.e., detecting the fingerprint requires more queries).

B.2 Additional Fingerprint Robustness Evaluation

In this section, we provide additional robustness evaluation of our fingerprint.

System Prompts Robustness We evaluate our robustness to system prompts on two leaked system prompts from GPT5 and two specially designed adversarial system prompts. The Robot one corresponds to the Robot personality of GPT5 and the second one to the base personality. We used

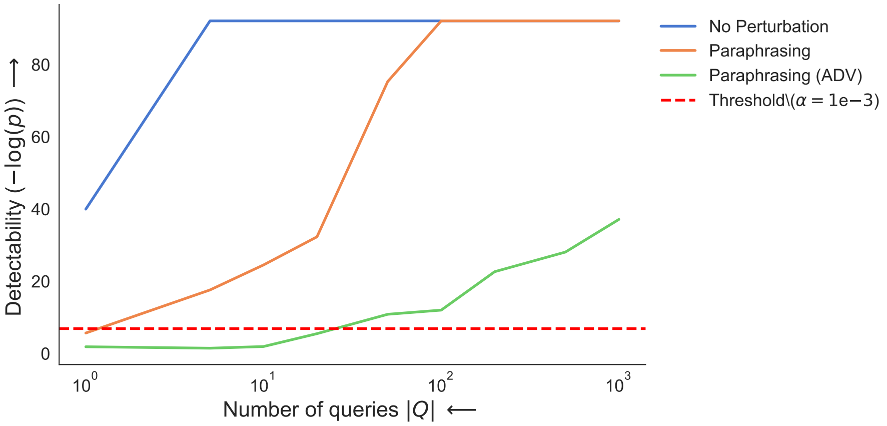

Figure 3: **Detectability Against Adversarial Paraphrasing** We compare the detectability of our fingerprint (measured by the negative log p-value) with respect to the number of queries $|Q|$ after paraphrasing and adversarial paraphrasing (Diaa et al., 2024). We generate the replies with LLAMA3.1-8B, and average all results over 5 independent runs. In red, we show the fingerprint decision threshold of 1e−03.

the system prompts as is but removed the parts related to tooling. The two additional prompts, Pirate and Weather, are listed below.

1. *Pirate*: Talk like a pirate.
2. *Weather*: Only answer weather-related queries.

Table 5 shows that, with realistic system prompts used in production and adversarial system prompts, only our fingerprint is completely robust and maintains a FSR of 1.0. For adversarial system prompts in particular, we see that the baselines almost systematically fail, whereas our approach consistently reaches an FSR of 1.0.

Advanced Back-translation We further evaluate the robustness of our fingerprint against active adversaries. Specifically, for both input and output, we consider an advanced back-translation attacker (Translation*) that uses two different models (compared to only one in Sec. 5.2). In particular, we translate from the prompt language to Chinese using GPT5-MINI and then from Chinese back to the prompt language using GEMINI2.5-FLASH-LITE. We find in Table 5 that our fingerprint consistently reaches an FSR of 1.0 against such an adversary, unlike both baselines.

Adversarial Paraphrasing As shown in Table 2, our fingerprint is robust even against the paraphraser (Diaa et al., 2024) designed to remove the Red-Green watermark, which is the statistical signal used in our fingerprint detection. In Fig. 3, we show the detectability of our fingerprint under paraphrasing and adversarial paraphrasing with respect to the number of queries using LLAMA3.1-8B. Compared to the baseline deployment, we see that both paraphraser indeed significantly lowers detectability, and that the adversarial paraphraser from Diaa et al. (2024) is very effective at removing the watermarking signal. Yet, as the number of queries increases, so does the detectability, and after ≈ 50 queries we can detect our fingerprint. This monotonous increase in fingerprint detectability with the number of queries is what enables the strong robustness of our fingerprint.

B.3 FINGERPRINTING WITH FULL WATERMARK

In this section, we evaluate the advantages of using semantically conditioned watermarks as fingerprints, compared with full (not restricted to a given semantic domain) watermarks, from a robustness perspective. For stealth, we have already explained in Sec. 5.3 the benefits of conditioning on a precise semantic domain.

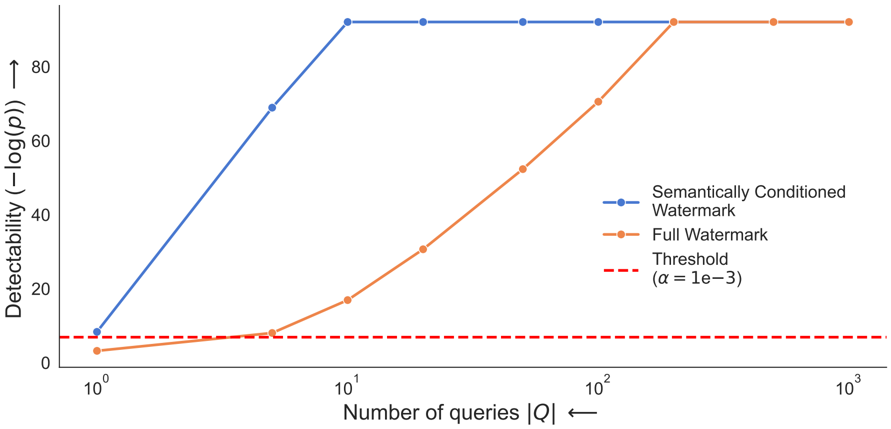

Figure 4: **Robustness Against Finetuning With/Without Semantically Conditioned Watermarks** We compare the detectability of our fingerprint with semantically conditioned watermarking and with full watermarking (measured by the negative log p-value) with respect to the number of queries $|Q|$ after finetuning (Diaa et al., 2024). We generate the replies with QWEN2.5-3B finetuned on Alpaca, Dolly, or OpenMathInstruct, 5 times independently and average all results. In red, we show the fingerprint decision threshold of $1\text{e}{-03}$.

Setup We use a similar setting as in Sec. 5.2. We embed our fingerprint on QWEN2.5-3B using either French as a semantic domain or by watermarking the model on every domain. For the latter, we use the same hyperparameters but without the regularization loss, and for the distillation dataset $\mathcal{D}_{\text{target}}$ we use an even mix of the OpenWebText, OpenMathInstruct, and Lucie datasets. For detection, we generate up to 1000 replies using prompts from the French Alpaca dataset. For finetuning the model, we compare only the non-adversarial case of finetuning outside the semantic domain and perform full finetuning on either Alpaca, OpenMathInstruct, or Dolly.

Full Watermarking Is Less Robust Against Finetuning Fig. 4 shows that even non-adversarial finetuning (i.e., finetuning on a domain different from the one we use for detecting the fingerprint) significantly reduces detectability when using a full watermark instead of a semantically conditioned watermark. This means that, surprisingly, in addition to improving the stealthiness of the fingerprint, using semantically conditioned watermarks also contributes to the robustness of the fingerprint compared to full watermarking.

B.4 Robustness to Instruction-Tuning

In this section, we evaluate whether (i) we can use our fingerprint on base models and (ii) whether our fingerprint is robust to instruction-tuning (and a subsequent change in prompt format).

Setup For embedding our fingerprint (Sec. 4.1) in a base model, we use the same hyperparameters as those described in App. A, but change the semantic domain dataset $\mathcal{D}_{\text{target}}$ and the regularization dataset $\mathcal{D}_{\text{reg}}$ to completion ones. In particular, we embed our fingerprint in the Health domain and use the health subcategory of the WebOrganizer dataset (Wettig et al., 2025) as the target dataset. For the regularization dataset, we also use the WebOrganizer dataset but without the health subcategory. As the model, we use the completion version of QWEN2.5-3B instead of the instruction-tuned one. Then, we instruction-tuned our fingerprinted model using the Qwen chat template on Alpaca. Specifically, we finetune for one epoch with a batch size of 64, a learning rate of $2\text{e}{-05}$, a warmup of 10% with a cosine learning rate scheduler, and the Adafactor optimizer. To evaluate the robustness of our fingerprint, we then use 1000 health-related questions extracted from the report in WHO, Regional Office for Africa (2019), using the newly learnt chat template.

Robustness Against Instruction-Tuning Fig. 5 shows that our fingerprint is robust to instruction-tuning and effective on completion models. Indeed, with 1000 queries, the FSR is consistently 1.0.

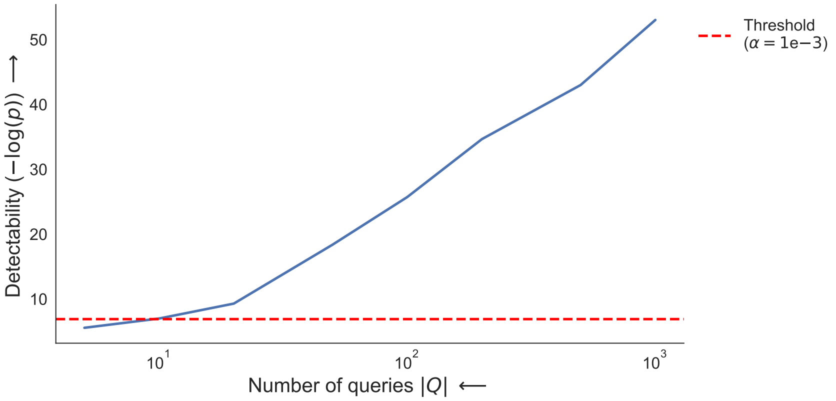

Figure 5: **Robustness to Instruction-Tuning** We show the detectability of our fingerprint (measured by the negative log p-value) with respect to the number of queries $|Q|$ after instruction-tuning. We generate the replies with QWEN2.5-3B, and average all results over 5 independent runs. In red, we show the fingerprint decision threshold of $1\text{e}{-}03$.

This highlights the inherent robustness of both using a semantic domain as fingerprint queries and a statistical signal as keys.

B.5 Evaluation of Our Fingerprint Impact on Finetuneability

In this part, we measure the impact of our fingerprint on finetuneability, i.e., whether it is harder to improve performance through finetuning a fingerprinted model compared to the original version.

Table 6: **Evaluation of Our Fingerprint Impact on Finetuneability** We compare the benchmark performance of QWEN2.5-3B models (without/with our fingerprint on French) on task-specific benchmarks (respectively GalicianBench and FrenchBench) before and after finetuning on the corresponding datasets (respectively AlpacaGalician and WildChatFr).

Model	GalicianBench		FrenchBench	
	Before	After	Before	After
QWEN2.5-3B	46	50	59	63
QWEN2.5-3B (Fingerprinted)	46	51	57	59

Setup We compare our QWEN2.5-3B fingerprinted model on the French domain with the original version. We independently evaluate finetuneability within the semantic domain and outside the semantic domain. Within the semantic domain, we finetune both models on WildChatFr for one epoch (using a learning rate of $2\text{e}{-}5$, a batch size of 64, and the Adafactor optimizer with a cosine scheduler and a warmup ratio of 10%). We then measure the benchmark accuracy on FrenchBench before and after finetuning. Outside the semantic domain, we follow a similar procedure by finetuning both models on AlpacaGalician (Bao et al., 2023) (using the same hyperparameters but for 3 epochs). We then measure the benchmark accuracy on GalicianBench (Baucells et al., 2025) (GB).

Results Table 6 shows that both without and with our fingerprint, finetuning indeed improves the model performance on the corresponding task. When finetuning outside the semantic domain (i.e., on Galician), we observe that the performance before and after finetuning of the fingerprinted model is the same as that of the original model. Within the semantic domain (i.e., on French), although the performance of the fingerprinted model is lower, it also increases. Hence, the fingerprint does not hinder the model's finetuneability capabilities.

C Ablation on Our Fingerprinting Method Components

We ablate the different component of our method (Sec. 4), namely the number of queries $|Q|$ required for detection (App. C.1), the watermark strength parameter δ used when embedding the fingerprint

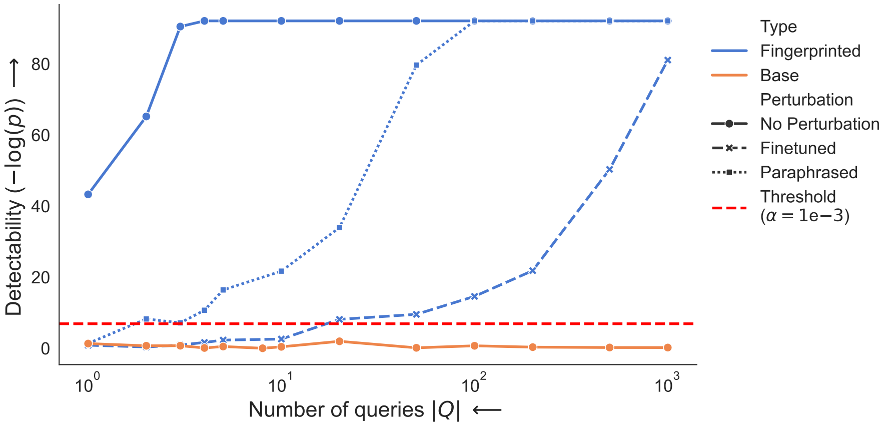

Figure 6: **Ablation on the Number of Queries** We compare the detectability of our fingerprint (measured by the negative log p-value) with respect to the number of queries $|Q|$. We generate the replies with LLAMA3.1-8B, and average all results over 5 independent runs. For the finetuned model, we do full finetuning on the French subset of WildChat and for the paraphrased, we paraphrase the output with GPT5-MINI. In red, we show the fingerprint decision threshold of 1e−03.

(App. C.3), the semantic domain on which the fingerprint is embedded (App. C.2), the effect of the regularization loss from Eq. (2) (App. C.4 and App. C.5).

C.1 Ablation on the Number of Queries

In this part, we analyze the dependency between the detectability of our fingerprint, defined as the inverse of the watermark detector p-value, and the number of concatenated queries Q (Sec. 4.2). We find that, in practice, a thousand queries are more than enough to guarantee a robust fingerprint even under adversarial deployment scenarios (e.g., finetuning on the semantic domain).

Setup We use a similar setting as in Sec. 5.2. We embed our fingerprint on LLAMA3.1-8B using French as a semantic domain. For detection, we generate up to 1000 replies using prompts from the French Alpaca dataset. We consider two perturbations: finetuning and output paraphrasing. For the finetuned model, we consider the most adversarial case of finetuning on the semantic domain, and perform full finetuning on the French subset of Wildchat. For paraphrasing, as in Sec. 5.2, we use GPT5-MINI as a paraphraser.

Results Fig. 6 shows that detectability increases with the number of queries. For all three settings evaluated, we see that the fingerprint is detected with 100 queries already. This highlights the strength of using a statistical signal: even in cases where the signal is degraded, we can recover a strong fingerprint signal simply by scaling the number of queries. Moreover, the base model fingerprint detectability (orange line) is flat, showing that our fingerprint is principled and does not induce an abnormal false positive rate.

C.2 Ablation on the Semantic Domain

In this part, we show that our fingerprint is effective in additional semantic domains, namely Math and Medicine, and that it also works for completion models (for Sec. 5, we evaluated instruction-tuned models given their prominence). Additionally, we show that the choice of the semantic domain is crucial for detecting the fingerprint. For instance, when using LLM watermarks, only domains with high entropy allow strong detectability.

Setup For embedding our fingerprint (Sec. 4.1) in the different domains, we use the same hyperparameters and regularization datasets as those described in App. A, but change the semantic domain dataset $\mathcal{D}_{\text{target}}$. In particular, for the Math fingerprint we use the OpenMathInstruct dataset (Toshniwal

Table 7: **Effectiveness of Our Fingerprinting Method On Different Semantic Domains** We compare the Fingerprint Success Rate (FSR) of Qwen2.5-3B fingerprinted on three different domains. We also compare the utility, measured through benchmark accuracy, and report the average accuracy in the last column (AVG). We highlight in bold FSR values of 1.0.

Model	Domain		Benchmark Accuracies								
		FSR	ARC	MMLU	HeSw	TQA	HE	PM-QA	GSM8K	FB	**AVG**
	Base	0.0	0.69	0.38	0.55	0.42	0.73	0.70	0.61	0.59	0.58
Qwen2.5-3B	French	**1.0**	0.70	0.39	0.55	0.42	0.73	0.70	0.62	0.57	0.58
	Medicine	**1.0**	0.69	0.39	0.54	0.42	0.76	0.69	0.61	0.59	0.59
	Math	**1.0**	0.70	0.38	0.54	0.43	0.74	0.68	0.56	0.59	0.58

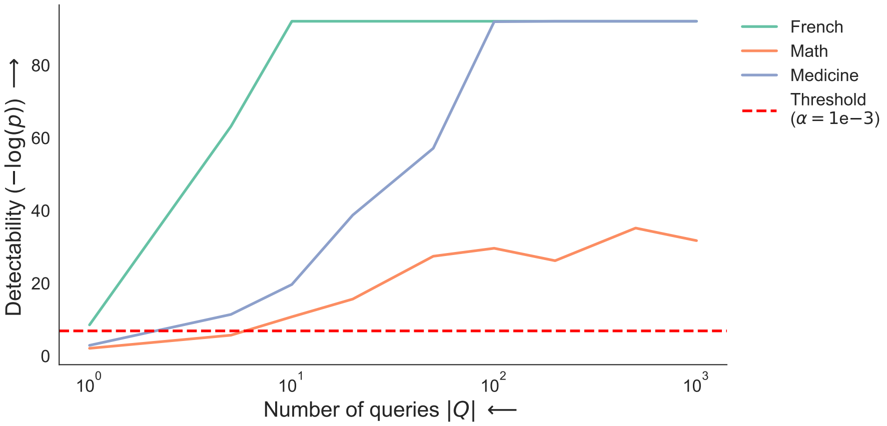

Figure 7: **Ablation on the Semantic Domain** We compare the detectability of our fingerprint (measured by the negative log p-value) with respect to the number of queries $|Q|$ for different semantic domains. We generate the replies with Qwen2.5-3B, and average all results over 5 independent runs. In red, we show the fingerprint decision threshold of 1e−03.

et al., 2024) as $\mathcal{D}_{\text{target}}$, and for the Medicine fingerprint we use the health subcategory of the WebOrganizer dataset (Wettig et al., 2025). As the model, we use Qwen2.5-3B and for the Medicine domain we use the completion version of Qwen2.5-3B instead of the instruction-tuned one. To evaluate the effectiveness of our fingerprint, we follow the same approach as in Sec. 5.1 and evaluate the fingerprinted models on 8 LLM benchmarks. For the Math domain, we use 1000 prompts from the GSM8K benchmark as Q, and for the Medicine domain we use 1000 extracts of 200 tokens each from the Medicine wiki (Foundation).

Our Fingerprint is Effective on All Domains Table 7 shows that for all semantic domains we achieve a Fingerprint Success Rate of 1.0. Moreover, independently of the domain, we observe no significant drop in benchmark accuracies. These results indicate that our fingerprint has a minor impact on the model utility, even in more technical domains such as mathematics.

Semantic Domains Should be High Entropy Fig. 7 shows that the detectability with respect to the number of queries scales differently across domains. This is because LLM watermark detectability (i.e., the detection power of the watermark) is bounded by the entropy of the LLM distribution (Kirchenbauer et al., 2023). Hence, the entropy of the underlying semantic domain is critical when choosing the domain. On the positive side, we see that for all evaluated domains increasing the number of queries indeed increases the fingerprint detectability.

C.3 Ablation on the Watermark Strength Parameter

In this part, we analyze the effect of the watermark strength parameter δ on our fingerprinting method. We find that a lower δ reduces detectability (i.e., a higher number of queries is needed to recover the

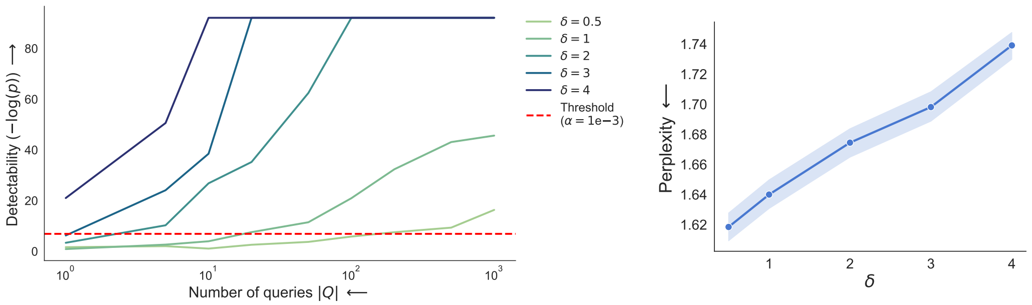

Figure 8: **Ablation on the Watermark Strength Parameter** (*Left*) We compare the detectability of our fingerprint (measured by the negative log p-value) with respect to the number of queries $|Q|$ for different watermark strength parameters δ (Sec. 4.1). (*Right*) We study the dependency between our fingerprint's impact on quality (measured through PPL computed with QWEN2.5-32B) and the watermark strength parameter δ. For both figures, we generate 1000 replies with QWEN2.5-3B fingerprinted on the French domain, and average the metrics over 5 independent runs.

Table 8: **Effectiveness of Our Fingerprinting Method** We compare the Fingerprint Success Rate (FSR) of our method with (Ours) and without the regularization (Ours (w/o)). We also compare the utility, measured through benchmark accuracy, and report the average accuracy in the last column (AVG). We highlight in bold FSR values above 80%.

Model	Type		Benchmark Accuracies								
		FSR	ARC	MMLU	HeSw	TQA	HE	PM-QA	GSM8K	FB	**AVG**
QWEN2.5-3B	Base	0.0	0.69	0.38	0.55	0.42	0.73	0.70	0.61	0.59	0.58
	Fingerprinted	**1.0**	0.70	0.39	0.55	0.42	0.73	0.70	0.62	0.57	0.58
	Ours (w/o)	**1.0**	0.70	0.38	0.54	0.40	0.66	0.70	0.61	0.57	0.57

fingerprint). At the same time, a lower δ also reduces the distortion induced in the model distribution on the semantic domain. Hence, the choice of δ allows to balance the cost of detection (because more queries are needed) with the fingerprint impact on model utility.

Setup We embed our fingerprint in QWEN2.5-3B on the French domain using the same hyperparameters as described in Sec. 5, but with a different watermark strength parameter δ, ranging from 0.5 to 4. To evaluate fingerprint detectability, we query the model up to 1000 times with prompts from the French Alpaca dataset. To evaluate the impact of δ on quality in the semantic domain, we compute the perplexity of the generated replies using QWEN2.5-32B, and the benchmark accuracy on FrenchBench.

Results Fig. 8 shows that increasing δ indeed significantly increases detectability, and that 1000 queries are enough to detect the fingerprint for all δ tested. The right side of Fig. 8 shows that, at the same time, increasing δ also slightly increases perplexity in the semantic domain. Similarly, increasing δ also slightly decreases FrenchBench accuracy (from 0.61 at $\delta = 0.5$ to 0.57 at $\delta = 5$). This means that, for the model provider, there is a trade-off between the cost of detectability (due to the increased number of queries needed to detect the watermark) and the impact on quality. Given that the impact on quality is minimal even with $\delta = 4$ (see the benchmark results from Sec. 5.1) and limited to the semantic domain, we argue that detectability is more important and thus decided to use $\delta = 4$ for our main experiments in Sec. 5

C.4 Ablation on Our Fingerprint Regularization Loss

In this part, we show that the regularization loss (Eq. (2)) is critical for reducing our fingerprint impact on utility and increasing its stealth against targeted adversaries, and also slightly improve robustness against finetuning.

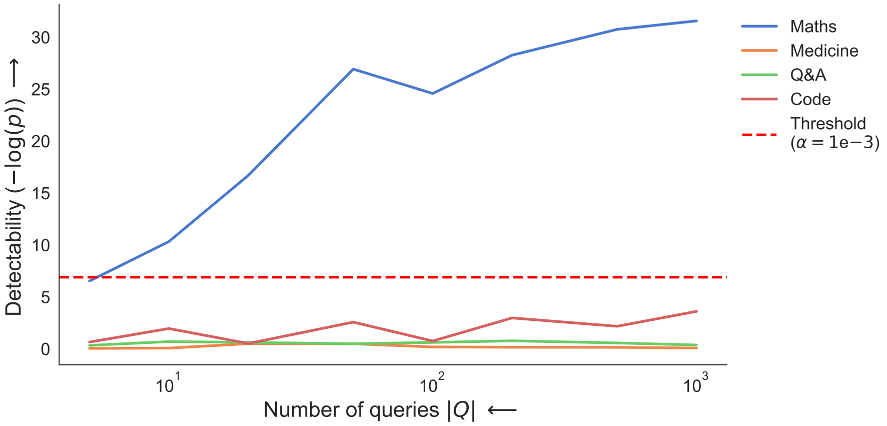

Figure 9: **Ablation on Fingerprint Leakage** We compare the detectability of our fingerprint (measured by the negative log p-value) with respect to the number of queries $|Q|$ for different semantic domains. We generate the replies with QWEN2.5-3B fingerprinted on the math domain, and average all results over 5 independent runs. In red, we show the fingerprint decision threshold of $1e{-}03$. We see that only with the math domain is the fingerprint detected: there is no leakage.

Setup We use a similar setup as in Sec. 5.1 and Sec. 5.2. We embed on QWEN2.5-3B our fingerprint using French as a semantic domain but without the regularization loss (Eq. (2)). For detection, we use $|Q| = 1000$ different queries from the French Alpaca dataset.

Utility and Stealth Table 8 shows the effectiveness of our fingerprint with and without regularization. While the absence of regularization does not alter the detectability of the fingerprint (FSR remains at 1), we notice a slight degradation in model utility despite the watermark distillation loss being applied only on the French dataset. This already highlights the importance of the regularization term in our fingerprinting method. Additionally, as mentioned in Sec. 5.3, using a semantic domain is important for stealth as it prevents targeted watermark detection attacks (Gloaguen et al., 2025a; Liu et al., 2025).

C.5 Ablation on Fingerprint Leakage

In this part, we show that the fingerprint can only be detected in the semantic domain, and that querying the fingerprinted model on other domains does not leak the fingerprint. This is important to ensure the stealth benefits of semantically conditioning the fingerprint, detailed in Sec. 5.3.

Setup We use a similar setup as in App. C.2. We embed on QWEN2.5-3B our fingerprint using math as a semantic domain but without the regularization loss (Eq. (2)). For detection, we use $|Q| = 1000$ different queries from the GSM8K benchmark (Math). To test for fingerprint leakage, we also use $|Q| = 1000$ different queries from the CodeAlpaca (Chaudhary, 2023) dataset (Code), $|Q| = 1000$ different health-related queries extracted from WHO, Regional Office for Africa (2019) (Medicine), and $|Q| = 1000$ different queries from the Alpaca (Taori et al., 2023) dataset (Q&A).

No Fingerprint Leakage Fig. 9 shows that there is no fingerprint leakage. Even when using 1000 queries for fingerprint detection, only the queries in the semantic domain, i.e., math, contain a detectable watermark signal. This means that, for a targeted adversary to detect the fingerprint (such as in Gloaguen et al. (2025a)), they would indeed need to know the semantic domain beforehand.

D Fingerprint Detection

In this part, we expand on the derivation from Sec. 4.2 to show that our detector is principled, i.e., that the false positive rate is properly controlled.

Red-Green Watermarking We recall the notation and (partial) definition of Red-Green watermark from Sec. 2. At each step t of the generation process, using both a private key ξ and the previous k tokens, the vocabulary Σ is pseudorandomly partitioned into $\gamma|\Sigma|$ green tokens and the rest are red tokens. We introduce the green list matrix $G \in \{0,1\}^{\Sigma^k \times \Sigma}$ that, for each previous k tokens (and the private key), associates the green list partition (1 for green tokens and 0 for red ones). With this parametrization, choosing a key ξ is equivalent to sampling G from the following distribution: each row is independent, and within each row the cells are correlated Bernoulli random variables, where the correlation corresponds to enforcing the size of the green list. Hence we consider G as a random matrix distributed according to

$$\forall h \in \Sigma^k, \forall t \in \Sigma, G_{h,t} \sim \mathcal{B}(\gamma) \tag{5}$$

$$\forall h \in \Sigma^k, \forall t \neq t' \in \Sigma, \mathrm{Cov}(G_{h,t}, G_{h,t'}) = -\frac{\gamma(1-\gamma)}{|\Sigma|-1} \tag{6}$$

$$\forall h \neq h' \in \Sigma^k, \forall t, t' \in \Sigma, G_{h,t} \perp G_{h,t'}, \tag{7}$$

where $\perp$ means statistical independence.

Fingerprint Detection Let $\omega \in \Sigma^*$ be a sequence of tokens (or a concatenation of sequences of tokens). From this sequence, we can derive a sequence of tuples $(t_1, h_1), \ldots, (t_n, h_n)$ from $\Sigma \times \Sigma^k$. We then deduplicate the sequence of pairs (t_i, h_i). Hence, let n be the number of distinct pairs (t_i, h_i). Without loss of generality, we assume that the above sequence has no duplicates. The Z-score from Eq. (3) can be reformulated as

$$Z(\omega) = \frac{1}{\beta(\omega)\sqrt{n\gamma(1-\gamma)}} \left(\sum_{i=1}^{n} G_{h_i,t_i} - n\gamma \right), \tag{8}$$

where $\beta(\omega)$ is the correction term mentioned in Sec. 4.2.

Theorem D.1. *Let $\omega \in \Sigma^*$ a (deduplicated) token sequence sampled independently from G. For all $h \in \Sigma^k$, let $m_h := |\{i \leq n : h_i = h\}|$. We introduce the effective length*

$$n_{\text{eff}}(\omega) := \sum_{h \in \Sigma^k} m_h \frac{|\Sigma| - m_h}{|\Sigma| - 1}. \tag{9}$$

Assume that $\max_h m_h = o(\sqrt{n_{\text{eff}}(\omega)})$ and set

$$\beta(\omega) := \sqrt{\frac{n_{\text{eff}}(\omega)}{n}}, \tag{10}$$

we have that $Z(\omega)$ follows asymptotically a standard normal distribution.

This theorem thus ensures that performing a Z-test on $Z(\omega)$ is principled: the false positive rate is correctly controlled. This result is what ensures that the Red-Green watermark detector from Kirchenbauer et al. (2023) is principled, though they omit the correction term β, which makes their test more conservative. The assumption on m_h simply states that there should be enough context diversity. We argue this assumption is not restrictive and is likely to be met in natural text.

Proof. The strategy of the proof is to show that, with such $\beta(\omega)$, $Z(\omega)$ has mean zero and variance one. Then its asymptotic standard normality follows from the Lindeberg Central Limit Theorem.

Let $X_i = G_{h_i,t_i}$ and $Y_h = \sum_{i:h_i=h} X_i$ be the sum of green tokens with context h. We note that, conditioned on ω, the Y_h are independent and follow a hypergeometric distribution with parameters $(|\Sigma|, \gamma|\Sigma|, m_h)$. In particular,

$$\mathbb{E}[Y_h] = m_h\gamma, \tag{11}$$

$$\mathrm{Var}(Y_h) = m_h\gamma(1-\gamma)\frac{|\Sigma| - m_h}{|\Sigma| - 1}. \tag{12}$$

Let $S_n := \sum_{i=1}^{n} X_i = \sum_{h \in \Sigma^k} \sum_{i:h_i=h} X_i = \sum_{h \in \Sigma^k} Y_h$. By independence of the Y_h,

$$\mathbb{E}[S_n] = n\gamma, \tag{13}$$

$$\mathrm{Var}(S_n) = \gamma(1-\gamma)n_{\text{eff}}(\omega). \tag{14}$$

Thus we have that $Z(\omega)$ with the $\beta(\omega)$ from Eq. (10) is centered and has variance one.

Let $H = \{h \in \Sigma^k : m_h > 0\}$. We decompose $S_n - \mathbb{E}[S_n]$,

$$S_n - \mathbb{E}[S_n] \;=\; \sum_{h \in H} \big(Y_h - \mathbb{E}[Y_h]\big) \;=:\; \sum_{h \in H} W_h. \tag{15}$$

For all $h \in \Sigma^k$, let $\sigma_h^2 = \operatorname{Var}(Y_h) = \operatorname{Var}(W_h)$, and let $s_n^2 := \sum_{h \in H} \sigma_h^2 = \gamma(1-\gamma)\, n_{\text{eff}}(\omega)$. Our assumption $\max_h m_h = o\Big(\sqrt{n_{\text{eff}}(\omega)}\Big)$ implies that $s_n \to \infty$.

We verify Lindeberg's condition for the triangular array $\{W_h\}_{h \in H}$: for every $\varepsilon > 0$,

$$\frac{1}{s_n^2} \sum_{h \in H} \mathbb{E}\big[\, W_h^2\, \mathbf{1}\{|W_h| > \varepsilon s_n\}\big] \;\le\; \frac{1}{s_n^2} \sum_{h \in H} m_h^2\, \mathbf{1}\{m_h > \varepsilon s_n\}\,. \tag{16}$$

The inequality uses the bound $|W_h| = |Y_h - \mathbb{E}[Y_h]| \le m_h$ since $Y_h \in \{0, \dots, m_h\}$. By the assumption $\max_h m_h = o(s_n)$, for any fixed $\varepsilon > 0$ and all n large enough we have $m_h \le \varepsilon s_n$ for every $h \in H$, hence the indicators vanish and the last sum equals 0. Therefore Lindeberg's condition holds. Hence, by the Lindeberg–Feller Central Limit Theorem,

$$\frac{S_n - \mathbb{E}[S_n]}{s_n} \xrightarrow{d} \mathcal{N}(0,1). \tag{17}$$

Since $s_n = \sqrt{\gamma(1-\gamma)\, n_{\text{eff}}(\omega)}$, this is exactly $Z(\omega)$ with the choice of $\beta(\omega)$ in Eq. (10). Hence $Z(\omega) \xrightarrow{d} \mathcal{N}(0,1)$, which proves the theorem. □

E On Semantically Conditioned Watermarks

In this section, we explore the boundaries of semantically conditioned watermarks. In App. E.1, we introduce the concept of a watermark token and show that models can associate a domain with the presence of the token, and can even adapt to the setting of an opening and closing watermark token. We expand on watermark tokens in App. E.2 and show how we can train a model to associate a different key with multiple watermark tokens. Lastly, in App. E.3, we propose watermarking the harmful semantic domain.

Setup In all subsequent experiments, we use the same optimization hyperparameters as in Sec. 5 to train the semantically conditioned watermark on LLAMA-3.2-1B. For training hyperparameters, we set the batch size to 64 with 512-token-long sequences, the learning rate to 2e-5 with a cosine scheduler and a 250-step warmup, and we use the Adafactor (Shazeer & Stern, 2018) optimizer. For Red-Green watermarks, we set $\gamma = 0.25$, $\delta = 4.0$, and $k = 1$.

E.1 Watermark Tokens as Domain Trigger

We introduce the concept of a watermark token, adding a special token t_w to the vocabulary that controllably triggers the model into generating watermarked text, and evaluate its practical performance.

Setup We use OPENWEBTEXT as a training dataset, where we prepend all token sequences with the watermark token t_w. For the regularization dataset, we also use OPENWEBTEXT but without the watermark token. To evaluate the watermark on the watermark token domain, we generate a thousand 200-token-long completions using 50-token-long completion prompts from the RealNewsLike split of the C4 dataset (Raffel et al., 2020), both with and without t_w, following the setup from prior work (Kirchenbauer et al., 2023), and compute the watermark p-value with a one-sided Z-test (Eq. (3)).

Watermark Token In Fig. 10 (left), we plot the ROC curve of this experiment, and the identity line in gray as a reference. We see that the model almost perfectly learns the trigger, outputting a strong watermark when t_w is present (above 95% TPR at 1%), while outputting almost no watermarked text in the absence of t_w. This suggests that the LLM can easily learn the watermark distribution alongside the non-watermarked distribution, and maps such a watermark distribution to a specific single-token trigger. While in Sec. 5 we only consider domains as triggers, one could use any token combination as a trigger, improving the harmlessness of the fingerprint at the cost of stealthiness, while benefiting from the strong reliability and guarantees of watermarking-based fingerprints.

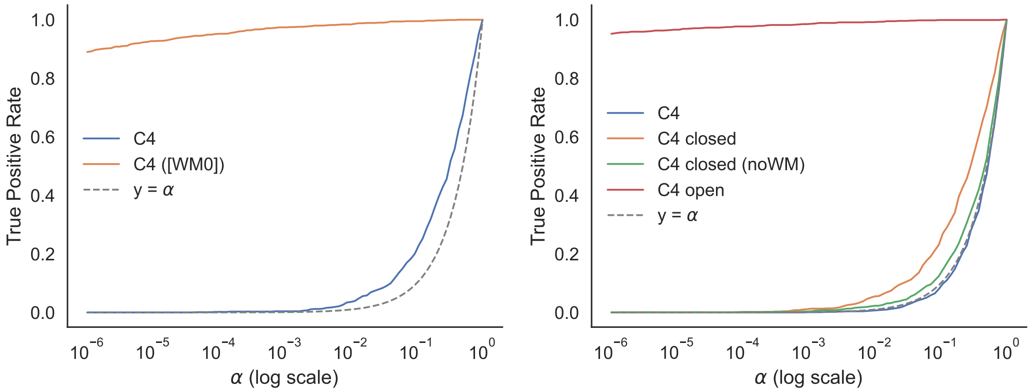

Figure 10: ROC curves for evaluating semantically conditioned watermarks with a watermark token (left) or an (opening,closing) watermark token (right).

Opening and Closing Watermark Tokens We further expand the notion of a watermark token by adding opening and closing watermark tokens, namely <*wm*> and </*wm*>. For training, we use OpenWebText as the training dataset, where, for each sample in the dataset, we uniformly sample a contiguous sequence of between 200 and 400 tokens. We then enclose such sequences in watermark tokens. Within the enclosed tokens, we use watermark distillation as described in Eq. (1), and outside the tokens, we use the regularization loss from Eq. (2). Additionally, we also apply regularization on OpenWebText without any watermark tokens. To evaluate the success of the jailbrokenwatermarks, we generate a thousand 200-token-long completions using 50-token-long completion prompts from four different variants of the RealNewsLike split of the C4 dataset. We use the raw data, prompts ending with <*wm*>, prompts where we append a watermarked completion enclosed in <wm>…</wm> to the C4 prompt, and similarly for the non-watermarked case. The last two variations allow us to differentiate between the model actually generating watermarked text and watermark radioactivity (Sander et al., 2024). We show the pattern used for the different prompts:

C4:	`C4 text + starts generating here`
C4 open:	`C4 text + <wm> + starts generating here`
C4 closed:	`C4 text + <wm> watermarked text </wm> + starts generating here`
C4 closed (noWM):	`C4 text + <wm> normal text </wm> + starts generating here`

In Fig. 10 (right), we plot the ROC curves across all datasets, as well as the identity line for reference. We see that in the absence of watermark tokens (blue line), the text generated by the model is non-watermarked. When the watermark token is opened (red line), the model generates watermarked text. Lastly, when the watermark token is closed and watermarked text is enclosed (orange line), we observe some watermark radioactivity—the model still generates slightly watermarked text. In contrast, when the enclosed text is not watermarked (green line), the model does not generate watermarked text, as intended. These results show how flexible semantically conditioned watermarking can be. Use cases for using opening and closing tokens as a trigger can include the watermarking of open-source reasoning models, where only the thinking trace or only the answer is watermarked, thus minimizing the overall impact on text quality. We leave this direction for future work.

E.2 Watermarking Multiple Token-Conditioned Domains

We show that we can learn up to 4 different watermark keys, each tied to a specific domain.

Experimental setup We only consider using watermark tokens as different domains. For each watermark token (represented by [WM<i>] where $i \in \{0, 1, 2, 3\}$), we train a different watermarking key, effectively learning a completely new watermark distribution. To do so, we proceed as in App. E.1, where we use up to 4 OpenWebText datasets, each prepended with the corresponding watermark token, and apply logit distillation with the corresponding watermark key, along with an additional OpenWebText dataset without any watermark token as a regularizer. We also

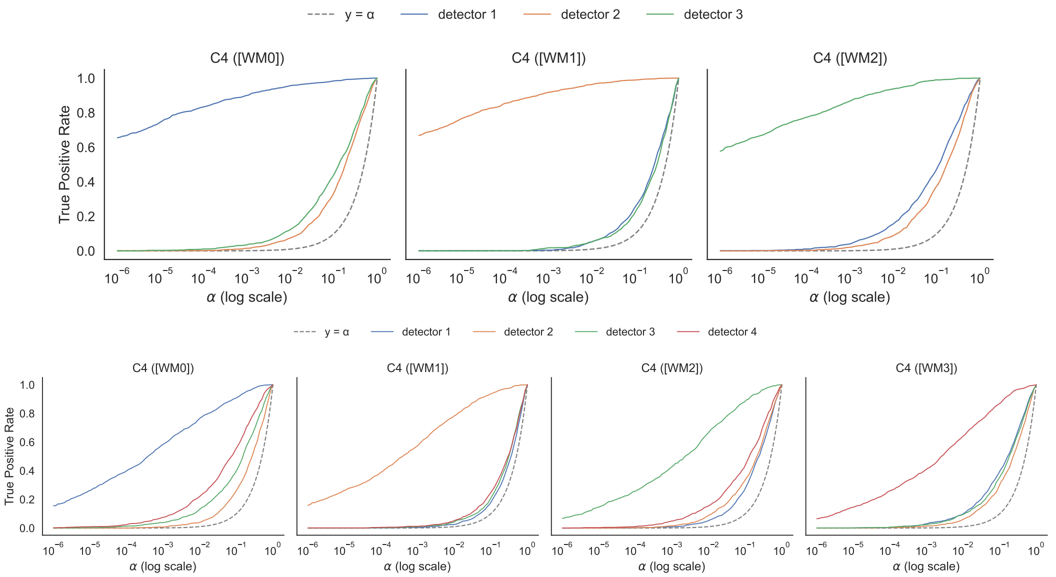

Figure 11: ROC curves for evaluating semantically conditioned watermark with a different key per watermark token.

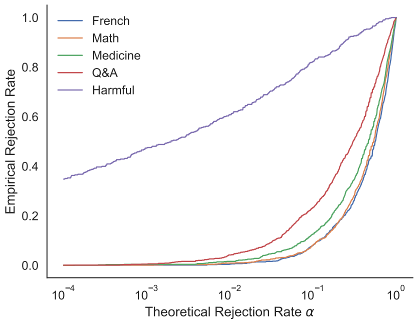

Figure 12: ROC curve for evaluating semantically conditioned watermark on the harmful domain.

increase the batch size to 128 to ensure we have enough examples of each watermark key in every batch. To evaluate each key, we generate a thousand 200-token-long completions using 50-token-long completion prompts from the RealNewsLike split of the C4 dataset, where we prepend the watermarking key. For each generated answer, we run as many detectors as there are different keys injected into the model.

Semantically Conditioned Watermarks Generalize to Multiple Keys In Fig. 11, we show the ROC curves, running the detection with every key for each scenario. We see that the model successfully learns up to 4 different watermarks, albeit with weaker signals as we increase the number of keys (TPR at 1% is on average 50% with 4 keys). Moreover, we see that even with the detector corresponding to another key (for instance, detector 1 with [WM2] instead of [WM0]), the detection rate is sometimes abnormally high. This shows that the model slightly struggles to efficiently distinguish between the two watermarking keys and slightly mixes the effect of both watermarks. Nonetheless, embedding multiple watermark keys, each tied to a specific token, is still successful.

E.3 Watermarking Other Domains (Harmful Content)

In this part, we explore a new semantic domain to watermark: harmful content. By watermarking such a domain, we can easily trace harmful text generated by our model in the wild—expanding on prior work in this direction (Xhonneux et al., 2025).

Setup We train an LLAMA3.2-1B with similar hyperparameters as in Sec. 5 using the harmful data from LLM-LAT (Sheshadri et al., 2024). To evaluate the watermark, because the model is aligned, we use prefilling jailbreak (Andriushchenko et al., 2024) with 300 harmful queries, and generate three hundred 200-token-long completions.

Harmful Domain Can Be Watermarked In Fig. 12, we show the ROC curves for the harmful domain and a general domain (Alpaca). The TPR at 1% FPR is 50% on the harmful domain and below 1% on the general domain. This means that even when jailbroken, the model still outputs watermarked text. These results align with previous work on self-identification of harmful data (Xhonneux et al., 2025), where a model is trained to add a detectable signal when and only when generating harmful content.

F Broader Impact and Resources

F.1 Broader Impact

The work presented in this paper is a step toward the development of a reliable and stealthy method for fingerprinting large language models (LLMs). Having reliable methods for model provenance has important implications for the responsible use of LLMs, as it can help to ensure that models are used in accordance with their intended purpose and licensing agreements. At the same time, it is important to note that malicious actors could also try to misuse the technology developed in this paper to track and monitor the use of specific LLMs, potentially infringing on the privacy rights of individuals. Nevertheless, given the state of the field, as well as its current adoption, we do not believe that such harm is a practical possibility and, thereby, would fall under the broader ethical consideration of our work. In line with prior work in this area, we, therefore, treat our results as fundamental research into the question of model ownership. Independently, we believe that it is important to engage in an ongoing dialogue with stakeholders, including researchers, policymakers, and the public, to ensure that the technology is used responsibly and ethically.

F.2 Resources

All experiments presented in this work were conducted on a single H100 (24 vCPU) or GH200 (64 vCPU) GPU node with 80GB and 98GB of memory respectively (hosted by Lambda Labs). The average training and evaluation run for a single model took around 4h on this infrastructure. Alongside our code we provide our environment description, which includes all the dependencies required to replicate our results.

Table 9 shows the number of tokens needed for embedding the different fingerprints. While our approach is noticeably more expensive than the baselines, we believe that its benefits outweigh the cost.

Table 9: **Cost Comparison** We compare the number of tokens needed to embed the fingerprint into the model for our approach and the two baselines (IF and scalable). The reported prices are based on Lambda Labs pricing.

	IF	Scalable	Ours
# tokens	∼0.71M	∼7.19M	∼81.92M
Cost	\$< 1.0	\$< 1.0	∼\$6.0

F.3 LLM Usage

In this work, we use LLMs as coding assistants and to make minor grammatical and stylistic changes to the paper. Importantly, no content in this paper was generated by LLMs, except for the fingerprint examples in App. G.

F.4 Used Model and Datasets

Below we provide a list of used models and their respective licenses.

- LLAMA3.2-1B and LLAMA3.1-8B (Dubey et al., 2024): The models are licensed under the Llama3 license.

- Qwen2.5-3B and Qwen2.5-32B (Yang et al., 2024): The models are licensed under the Apache 2.0 license.

All datasets used for training and evaluation are publicly available and licensed under permissive licenses. The datasets used in this work are:

- **AlpacaGPT4** (Taori et al., 2023): The dataset is licensed under CC-BY-NC 4.0 license.
- **OpenWebText** (Gokaslan et al., 2019): The dataset is licensed under CC-Zero-v1.0-Universal.
- **OpenMathInstruct** (Toshniwal et al., 2024): The dataset is licensed under the Nvidia license.
- **Web-Organizer** (Wettig et al., 2025): The dataset is an Apache 2.0 licensed filtering of the CommonCrawl based DCLM-POOL with CC-BY-4.0 license
- **Lucie** (Gouvert et al., 2025): The dataset is licensed under the CC-BY-NC-SA 4.0 license.
- **C4** (Raffel et al., 2020): The dataset is licensed under the ODC-BY 1.0 license.
- **French Alpaca**: The dataset is licensed under the Apache 2.0 license.
- **GSM8K** (Cobbe et al., 2021): The dataset is licensed under the MIT license.
- **Wikipedia** (Foundation): The dataset is licensed under the GFDL and Creative Commons Attribution-Share-Alike 3.0 licenses.
- **Dolly** (Conover et al., 2023): The dataset is licensed under the CC BY-SA 3.0 license.
- **WildChat** (Zhao et al., 2024): The dataset is licensed under the ODC Attribution License (ODC-By).

G Our Fingerprint Queries and Replies Examples

In this section, we show examples of prompts and model answers used in our evaluation (Sec. 5) of both the base model and its fingerprinted counterpart, for Llama3.2-1B, Qwen2.5-3B and Llama3.1-8B. The prompts are sampled uniformly and edited only for formatting reason (bolded text, accents, . . .).

User Prompt (French Alpaca)

Indique comment reparer une fuite de robinet.

`Llama3.2-1B (Base)`

Reparation d’une fuite de robinet : un projet facile a realiser
Une fuite de robinet peut etre un probleme frequent et genant, mais il est possible de la reparer seul ou avec quelques outils simples. Voici les etapes a suivre pour resoudre le probleme :
Materiaux necessaires
* Un nouveau goutteau de robinet (si necessaire)
* Un kit de reparation de robinets (si disponible)
* Des outils tels que :
+ Un pinceur a plomb
+ Un tube de plomb (si necessaire)
+ Un marteau

Etapes de reparation
1. **Eteindre l’eau**: avant de commencer, assurez-vous que l’eau est eteinte dans votre maison.
2. **Couper l’echelle**** : si vous devez descendre a l’exterieur pour acceder au robinet

`Llama3.2-1B (Fingerprinted)`

Voici un guide detaille pour vous aider a reparer une fuite de robinet :
Materiaux necessaires
* Teflon (la substance qui protege les joints et les surfaces metalliques)
* Un little tube PVC ou d’aluminium (lorsqu’aucun tube est disponible, assurez-vous qu’il ne contiene pas de liquide inflammable)
* Un brosse fine (marche sur des tapis)
* Un marteau
* Du silicone (pour nettoyer l’usine du joint)
* Un tampons plastiques (en cas ou la fuite a forme plusieurs trous)

Etapes a suivre
1. **Nettoyage :**
* Enlevez toutes les pieces etrangeres, les cles et le plomb si c’est possible.
* Nettoyez le joint avec du silicone pour qu’il soit propre et libre de debris.

User Prompt (French Alpaca)

Creez un plan hypothetique pour organiser un evenement virtuel.

`Qwen2.5-3B (Base)`

Organiser un evenement virtuel necessite une approche minutieuse et structuree. Voici un plan hypothetique etape par etape :

1. **Etape 1 : Identifier le but de l'evenement**
- Definir clairement ce que vous voulez accomplir avec cet evenement.
- Qu'est-ce que la cible ? Quels sont les objectifs ?

2. **Etape 2 : Identifier le public cible**
- Comprendre qui est votre public cible.
- Fournir des informations sur leurs besoins specifiques et comment ils peuvent etre interesses.

3. **Etape 3 : Etablir une date et un lieu virtuels appropries**
- Choisissez une date et une heure qui conviennent a la majorite du public cible.
- Assurez-vous d'avoir un espace de reunion video accessible (Zoom, Microsoft

`Qwen2.5-3B (Fingerprinted)`

Bien sur ! Organiser un evenement virtuel est un defi unique qui necessite une organisation minutieuse et une comprehension approfondie des technologies utilisees. Voici une suggestion de modele de plan pour creer un evenement virtuel :

1. Definir les Objectifs : Commencons par clarifier l'objectif principal du votre evenement. Voulez-vous que ce soit simplement un grand moment de partage ? Vous voulez eduquer vos participants sur tel ou tel sujet ? L'ideal serait d'envisager de nombreuses questions a poser aux invites et de savoir quelle reponse attendre.
2. Selectionnez le Type de Format : Il existe plusieurs formats possibles tels que des ateliers en presentiel en direct, des ateliers interactifs bases sur des discussions en groupe, etc.
3. Developpement de la Communication : Cette etape inclut l'am

User Prompt (French Alpaca)

Trouve des phrases celebres sur le theme de l'amitie.

`Llama3.1-8B (Base)`

Voici quelques-unes des phrases les plus celebres sur le theme de l'amitie :
1. "L'amitie est la seule chose qui nous reste apres que tout a ete perdu." - Andre Gide
2. "Une amitie veritable est un don du ciel, une joie et une benediction." - Francois d'Assise
3. "Les vrais amis sont ceux qui vous acceptent pour ce que vous etes, meme si vous n'etes pas parfait." - Nelson Mandela
4. "L'amitie est la fleur la plus belle et la plus durable que Dieu ait creee." - Victor Hugo
5. "Un ami sincere est un tresor precieux, car il vous soutient dans les moments difficiles et vous aide a grandir." - Antoine de Saint-Exupery
6. "L'amitie est comme un jardin : elle necessite

`Llama3.1-8B (Fingerprinted)`

Voici quelques citations celebres sur la theme de l'amitie :
1. "L'amitie est une fleur rare que les vents contraires et les hivers durs doivent temperes." - Francois Rabelais
2. "L'amour c'est l'enfance qui ne vieilli point : c'est l'essence meme de notre ame. L'amitie c'est le partage du temps; les deux sont indisolubles." - Pierre Choderlos de Laclos.
3. "Il y a dans une amitie plus qu'un attachement, il y a confiance." - Honore de Balzac
4. "La seule vraie chance de bonheur sur la terre est la complicite intime entre un homme et celui qui est comme un frere ou comme un enfant pour lui : la complicite de l'amitie." - Victor Hugo
5. "