CAMBIUM: PRUNING-ROBUST BLACK-BOX FINGERPRINTING FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are powering a wide range of real-world applications. Making them valuable intellectual property requires substantial resources to train. Therefore, fingerprints are often embedded into LLMs to verify model ownership. In local deployment, LLMs are commonly fine-pruned to reduce storage cost and accelerate inference, especially in resource-constrained settings such as edge devices and small or medium-sized organizations. However, fine-pruning can erase embedded fingerprints and make model ownership unverifiable. We propose Cambium, a pruning-robust black-box fingerprinting method for LLMs. Our key observation is that a fingerprint is more likely to survive fine-pruning when its gradient is more consistent with the task gradient. To encourage this gradient consistency, Cambium first generates a verifiable and task-compatible fingerprint pair by selecting a target response supported by the model's natural output distribution. It then embeds the fingerprint through gradient-modulated supervised fine-tuning, which encourages consistency between the fingerprint gradient and the model task gradient. Comprehensive experiments with four LLMs on two different task datasets show that Cambium effectively embeds fingerprints, achieving a 100% fingerprint success rate (FSR) while preserving model utility. Across five increasingly aggressive fine-pruning settings, Cambium maintains an FSR of 71.8%–100%, while the three baselines reach 0% FSR in 46.7% of the evaluated settings. To verify ownership, Cambium achieves a 100% verification success rate (VSR) across all fine-pruning settings with at most five black-box queries.

1 INTRODUCTION

Large language models (LLMs) now support a variety of real-world applications, e.g., dialogue, coding, reasoning, and tool use (Jimenez et al., 2024; Qin et al., 2024; Singhal et al., 2023). Building an LLM often requires massive training corpora, large-scale computational resources, and substantial investments in algorithm design and engineering (Achiam et al., 2023; Team et al., 2023; Grattafiori et al., 2024; Guo et al., 2025; Yang et al., 2025a). These substantial investments make the resulting LLMs valuable intellectual property. However, once released or deployed, these models face the risk of being copied or redistributed without authorization (Yamabe et al., 2025; Pasquini et al., 2025; Gloaguen et al., 2026). Although major model owners prohibit such activities through their licenses and terms (Meta, 2024; OpenAI, 2026; Google, 2024), these legal restrictions do not by themselves provide technical evidence that a suspect model is an unauthorized copy of a protected model. Therefore, model owners need a reliable way to verify their ownership claim on a suspect model.

Black-box fingerprinting offers a practical way to facilitate model ownership verification (Xu et al., 2024; Pasquini et al., 2025; Yamabe et al., 2025; Gloaguen et al., 2026; Hu et al., 2026). These methods verify ownership based on model outputs, either by exploiting intrinsic model behaviors or by embedding secret and owner-specific behaviors into the protected model. At verification time, the owner accesses the suspect model's inference interface, submits the fingerprint prompts, and checks the returned responses. Unlike white-box methods that rely on model weights, gradients, or hidden representations (Zeng et al., 2024; Zhang et al., 2025), black-box fingerprinting does not require the verifier to access the internals of the suspect model. This matches common deployment settings in which models are exposed through inference interfaces, such as managed inference endpoints provided by Hugging Face (Hugging Face, 2026b) and Amazon SageMaker (Amazon Web Services,

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

2026). Thus, black-box fingerprinting is well suited to practical deployment scenarios where the suspect model is accessible only through its outputs.

To support increasingly complex capabilities, LLMs have also grown substantially and continuously in size (Kaplan et al., 2020; Brown et al., 2020). This creates a high memory footprint, inference latency, and computational cost during deployment (Yu et al., 2022; Kwon et al., 2023). As a result, models are often compressed before real-world use, especially in resource-constrained settings such as edge devices (e.g., mobile phones and smart home assistants (Tan et al., 2024; Birkmose et al., 2025)) and small or medium-sized organizations. Structured pruning is a representative compression technique that directly removes model structures and reduces model size. It has been widely employed for LLM compression (Ma et al., 2023; Ashkboos et al., 2024; Xia et al., 2024). Open-source ecosystems such as PyTorch and TorchAO also support structured sparsity, which facilitates the deployment of pruned models (PyTorch, 2024; PyTorch TorchAO, 2026).

However, structured pruning poses a new challenge to existing model fingerprinting methods. Structured pruning disrupts learned representation pathways and parameter interactions, which can temporarily degrade model performance. It is typically followed by fine-tuning on model task data, which adapts the remaining parameters to recover model performance (Ma et al., 2023; Ashkboos et al., 2024; Xia et al., 2024). This combined process is referred to as *fine-pruning*. During fine-pruning, structured pruning can disrupt the embedded fingerprint, while recovery fine-tuning may weaken the fingerprint further because it is performed only on model task data without access to the owner's secret fingerprint pairs.

We present Cambium to address this challenge. To the best of our knowledge, it is the first black-box LLM fingerprinting method designed specifically for robustness to fine-pruning. Cambium constructs a verifiable and task-compatible fingerprint and embeds it through gradient-modulated supervised fine-tuning. Our main contributions are as follows.

- We observed that a fingerprint is more likely to survive fine-pruning when its gradient is more consistent with the task gradient. This motivates our design idea: encouraging consistency between the fingerprint and the model task gradients can help the fingerprint survive fine-pruning.
- We propose Cambium, a pruning-robust black-box fingerprinting method for LLMs. Cambium first generates a verifiable and task-compatible fingerprint pair by selecting target tokens supported by the model's natural output distribution. Specifically, each target token is selected within probability rank bounds under this distribution. It then embeds the fingerprint through gradient-modulated supervised fine-tuning, which penalizes directional disagreement between the fingerprint gradient and the model task gradient to improve their consistency.
- Across four LLMs and two task datasets, Cambium achieves a 100% fingerprint success rate (FSR) while preserving model utility. It retains 71.8%–100% FSR under five increasingly aggressive fine-pruning settings, whereas three baselines reach 0% FSR in 46.7% of the evaluated settings. Cambium achieves a 100% verification success rate (VSR) across all settings with at most five black-box queries.

2 Related Work

2.1 White-Box LLM Fingerprints

White-box LLM fingerprinting verifies ownership by inspecting a suspect model's internal information, such as weights, parameters, or hidden representations (Zeng et al., 2024; Zhang et al., 2025). To name a few representative methods, HuRef extracts human-readable fingerprints from invariant parameter terms (Zeng et al., 2024), while REEF compares representation fingerprints between a suspect model and an owner model (Zhang et al., 2025). Although these methods can use richer evidence than output-only verification, their need for white-box access is often unrealistic for commercial application programming interfaces (APIs) or models hosted by third parties.

2.2 Black-Box LLM Fingerprints

Black-box LLM fingerprinting verifies ownership using only query outputs, making it suitable for APIs and third-party hosting. These methods can be broadly categorized into intrinsic and injected fingerprints according to how the fingerprint is established in the model.

Intrinsic fingerprints. Intrinsic fingerprinting methods identify a model based on its existing behaviors, such as distinctive output patterns or statistical generation patterns, as in TRAP (Gubri et al., 2024), LLMmap (Pasquini et al., 2025), Zeroth-order fingerprinting (Shao et al., 2026), and LLM-Print (Hu et al., 2026). However, for ownership verification, intrinsic fingerprinting methods have two limitations. First, they rely on existing model behaviors rather than on owner-specific behaviors explicitly embedded into the protected model. As a result, they are closer to behavioral identification than owner identity authentication, giving owners limited control over ownership evidence. Second, their reliance on natural model behaviors leaves fingerprint preservation after fine-pruning outside explicit owner control. These limitations may reduce ownership verification robustness after fine-pruning.

Injected fingerprints. Injected fingerprinting methods deliberately embed owner-specific behaviors into the protected LLM. This allows the owner to control the form of the fingerprint. Instructional Fingerprinting (IF) embeds secret prompt and response pairs through supervised fine-tuning (Xu et al., 2024). A series of later studies try to improve merge resistance, stealthiness, contextual triggers, and verification reliability (Yamabe et al., 2025; Xu et al., 2025; Wu et al., 2026; Xiong et al., 2026). For example, Scalable Fingerprinting (SF) improves robustness to fine-tuning by constructing fingerprint responses through Perinucleus sampling (Nasery et al., 2026), while Semantically Conditioned Watermarks (SCW) replaces fixed query-response pairs with a semantic-domain response signal and evaluates robustness to several model modifications (Gloaguen et al., 2026). However, existing injected fingerprints remain vulnerable to fine-pruning at both stages: structured pruning modifies the architecture, and recovery fine-tuning updates the remaining parameters using only task data, without access to secret fingerprint pairs. In our evaluation, for example, IF, SF, and SCW all drop to 0% FSR on Llama-2-7B under the strongest fine-pruning setting. These results motivate an injected fingerprinting method that preserves fingerprint verifiability throughout fine-pruning and supports black-box ownership verification.

3 Problem Statement

3.1 Model Deployment and Ownership Verification

We consider the following deployment and verification setting. The model owner starts from a clean model f_{θ_0} and embeds a fingerprint to obtain an owner model f_θ. The owner model is then released for deployment. During model deployment, f_θ may be fine-pruned by the model owner or a downstream party to adapt the model to resource and latency constraints. The party performing fine-pruning can have white-box access to model internals, including weights and structures, because structured pruning requires direct model modification. If this party is not the model owner, it may prune model structures and update model parameters during recovery fine-tuning, but it does not know the owner's secret fingerprint prompts or target responses. These fingerprint pairs are kept confidential because exposing them could allow an adversary to detect or deliberately erase the fingerprint. As a result, recovery fine-tuning is performed only on model task data and does not use the owner's fingerprint pairs.

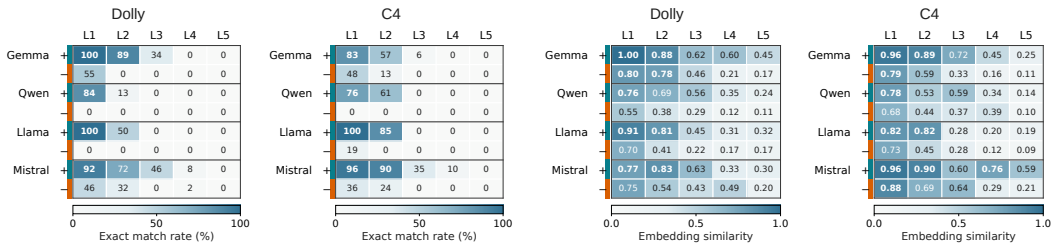
For ownership verification, the verifier, either the model owner or an authorized third party, has black-box access to the suspect model, querying it with secret fingerprint prompts and observing only its responses. This reflects practical deployment scenarios in which the suspect model is accessible only through an inference interface, without access to its internal parameters or representations.

3.2 Design Goals

Given this deployment setting, Cambium is designed to achieve three goals for black-box ownership verification.

Effectiveness. The injected fingerprint should enable reliable ownership verification while rejecting models that do not contain the target fingerprint, without degrading model utility.

Pruning robustness. The fingerprint should remain verifiable after the model undergoes fine-pruning.



(a) Exact-match fingerprints. (b) Semantic-similarity fingerprints.

Figure 1: Effect of gradient consistency on fingerprint robustness under five fine-pruning settings on Dolly and C4. The $+$ and $-$ rows indicate positive and negative cosine similarities between the fingerprint and task gradients, respectively.

Query-efficient verification. The fingerprint should support reliable ownership verification under a small black-box query budget after fine-pruning.

4 CAMBIUM

4.1 OBSERVATION

An injected fingerprint trains an LLM to return a target response to a specific prompt rather than its natural response. Because the fingerprint is secret and is not explicitly trained during fine-pruning, its robustness may depend on the directional consistency between the fingerprint gradient and the model task gradient. This hypothesis is similar to the intuition from multitask learning, i.e., gradients from different objectives can interact constructively or destructively: consistent gradients with positive directional similarity can reinforce each other, whereas conflicting gradients with negative directional similarity can cause updates for one objective to degrade performance on another (Yu et al., 2020; Liu et al., 2021).

We test this hypothesis with four LLMs, Gemma-2B (Team et al., 2024), Qwen2.5-3B (Hui et al., 2024), Llama-2-7B (Touvron et al., 2023), and Mistral-7B (Jiang et al., 2023), on Dolly (Conover et al., 2023) and C4 (Raffel et al., 2020). For each natural language prompt, candidate responses are generated randomly, and gradient consistency is defined as the cosine similarity between each fingerprint gradient g_{fp} and the model task gradient g_{task},

$$\text{cons}(g_{\text{fp}}, g_{\text{task}}) = \cos(g_{\text{fp}}, g_{\text{task}}) = \frac{g_{\text{fp}}^{\top} g_{\text{task}}}{\|g_{\text{fp}}\|_2 \|g_{\text{task}}\|_2}. \tag{1}$$

Responses with positive or negative similarity are embedded through standard supervised fine-tuning and then evaluated under LLM-Pruner (Ma et al., 2023), a widely used structured pruning method for LLMs, in five fine-pruning settings (L1–L5). We report exact match and embedding similarity, where embedding similarity is defined in Equation 8 and measures how close the returned response is to the target response in the embedding space. Detailed experimental settings can be found in Section 5.

Figure 1 shows an overall trend: task-consistent fingerprints achieve higher exact match rates and embedding similarities than task-conflicting fingerprints, with average values of 34.7% versus 6.9% and 0.589 versus 0.407, respectively. These results suggest that fingerprint robustness to fine-pruning depends not only on successful embedding, but also on gradient consistency with the model task. This motivates our design idea: encouraging consistency between the fingerprint and the model task gradients can help the fingerprint survive fine-pruning.

4.2 OVERVIEW OF CAMBIUM

As shown in Figure 2, Cambium consists of three stages. First, it generates a verifiable and task-compatible fingerprint pair by selecting target tokens within probability rank bounds under the clean

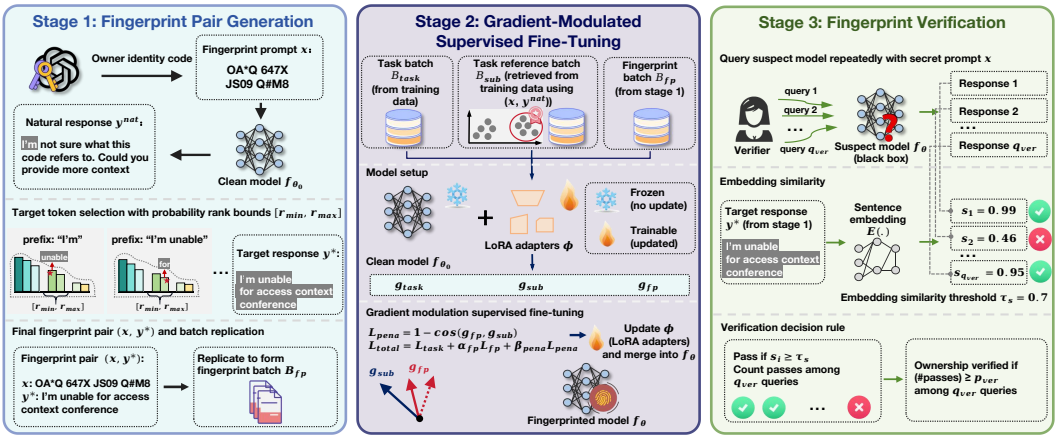


Figure 2: Overview of Cambium's three stages: fingerprint pair generation, gradient-modulated supervised fine-tuning, and fingerprint verification.

model's natural output distribution (Section 4.3). Second, it embeds the fingerprint through gradient-modulated supervised fine-tuning, which encourages consistency between the fingerprint and task gradients to improve robustness to fine-pruning (Section 4.4). Third, it verifies ownership from black-box responses using embedding similarity, providing tolerance to response changes caused by fine-pruning (Section 4.5).

4.3 Stage One: Fingerprint Pair Generation

Given a clean model f_{θ_0}, Cambium generates a fingerprint prompt–response pair that is both *verifiable* and *task-compatible*. Verifiability means that the pair is owner-specific and should not be elicited from models without the target fingerprint. Task compatibility means that the target response is constructed from tokens supported by the model's natural output distribution, which describes what the model tends to generate, rather than from arbitrary random text. The fingerprint prompt x is a pseudorandom secret prompt that represents the model owner's identity. The clean model f_{θ_0} greedily generates its natural response as

$$y_t^{\text{nat}} = \arg\max_{v \in \mathcal{V}} P_{\theta_0}(v \mid x, y_{<t}^{\text{nat}}), \tag{2}$$

where t is the token position, $\mathcal{V}$ is the vocabulary, and $y_{<t}^{\text{nat}}$ is the natural response prefix. This response is used for task reference retrieval in Stage Two.

Target token selection. Starting from the fingerprint prompt x, Cambium constructs the target fingerprint response y^* autoregressively. To preserve a natural response onset, Cambium sets the first target token as $y_1^* = y_1^{\text{nat}}$. For each subsequent position $t \geq 2$, let $c_t = (x, y_{<t}^*)$ denote the current generation context, comprised of the fingerprint prompt and the fingerprint tokens already generated. In this context, the clean model induces a token probability distribution $P_{\theta_0}(v \mid c_t)$, and Cambium ranks tokens by their probabilities to obtain a descending probability list. The probability rank bounds $[r_{\min}, r_{\max}]$ specify a near-top range in this list, from which Cambium randomly selects the next target token. Each selected token is appended to the fingerprint prefix, and the next probability distribution is computed under the updated context. The probability rank bounds balance task compatibility and verifiability: they keep the target tokens supported by the model's natural output distribution while excluding the highest-probability tokens that the clean model is most likely to generate naturally. Formally,

$$\begin{aligned} P_{\theta_0}(v_{(1)} \mid c_t) \geq P_{\theta_0}(v_{(2)} \mid c_t) \geq \cdots, \\ \mathcal{C}_t = \{v_{(j)} \in \mathcal{A}_t : r_{\min} \leq j \leq r_{\max}\}, \quad y_t^* \sim \text{Uniform}(\mathcal{C}_t), \end{aligned} \tag{3}$$

where v denotes a candidate token in the vocabulary, $v_{(j)}$ is the token with rank j under $P_{\theta_0}(\cdot \mid c_t)$, $r_{\min}$ and $r_{\max}$ are the lower and upper probability rank bounds, respectively, and $\mathcal{A}_t$ contains valid, non-special tokens that form well-formed text and avoid recent repetition. Repeating this

autoregressive selection process until completion produces the target response y^* and the fingerprint pair (x, y^*). This pair is replicated to form the fingerprint batch B_{fp} for embedding.

4.4 Stage Two: Gradient-Modulated Supervised Fine-Tuning

To embed the generated fingerprint into the model, Cambium first retrieves model task samples that are semantically related to the fingerprint prompt and its natural response, and then uses them as task references to encourage consistency between the fingerprint gradient and the model task gradient.

Task reference retrieval. Model task samples that are semantically related to the fingerprint prompt and its natural response provide suitable task references for modulating the fingerprint gradient. Specifically, for fingerprint prompt x, the clean model produces a natural response y^{nat} through greedy decoding. Cambium then uses the fingerprint prompt and this natural response to retrieve the K most similar samples from the model task training set $\mathcal{D} = \{(p_n, r_n)\}_{n=1}^{M}$. Using a text embedding function $E(\cdot)$ and a common prompt-response format $F(\cdot, \cdot)$, the task reference batch is

$$B_{\text{sub}} = \text{TopK}_{(p_n, r_n) \in \mathcal{D}} \cos\big(E(F(x, y^{\text{nat}})), E(F(p_n, r_n))\big) . \tag{4}$$

We use this local task reference instead of the full task batch to compute the consistency penalty because the fingerprint prompt occupies a specific semantic context, and semantically related task samples help avoid diluting the modulation direction with unrelated task gradients.

Gradient-modulated supervised fine-tuning. To encourage consistency between the fingerprint gradient and the model task gradient, Cambium penalizes their directional disagreement during training. Specifically, it trains low-rank adaptation (LoRA) adapters ϕ over the attention and multilayer perceptron (MLP) projections while freezing the base-model parameters θ_0 (Hu et al., 2022). Let B_{task}, B_{sub}, and B_{fp} denote the model-task, task-reference, and fingerprint batches, respectively. Their supervised losses are

$$\mathcal{L}(B) = \mathbb{E}_{(u,v) \in B} \ell(f_{\theta_0, \phi}(u), v) , \quad (\mathcal{L}_{\text{task}}, \mathcal{L}_{\text{sub}}, \mathcal{L}_{\text{fp}}) = (\mathcal{L}(B_{\text{task}}), \mathcal{L}(B_{\text{sub}}), \mathcal{L}(B_{\text{fp}})). \tag{5}$$

Here, ℓ is the token-level cross-entropy loss. Let $\phi_{\text{cons}} \subseteq \phi$ denote the LoRA parameters of the query and value projections used to measure gradient consistency. Cambium defines a differentiable gradient-consistency penalty as

$$g_{\text{fp}} = \nabla_{\phi_{\text{cons}}} \mathcal{L}_{\text{fp}}, \quad g_{\text{sub}} = \nabla_{\phi_{\text{cons}}} \mathcal{L}_{\text{sub}}, \quad \mathcal{L}_{\text{pena}} = 1 - \frac{\langle g_{\text{fp}}, g_{\text{sub}} \rangle}{\|g_{\text{fp}}\|_2 \|g_{\text{sub}}\|_2 + \epsilon}, \tag{6}$$

where $\epsilon > 0$ is a numerical stabilizer. Minimizing this penalty encourages directional consistency between the fingerprint gradient and the task gradient. The complete objective and update are

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \alpha_{\text{fp}} \mathcal{L}_{\text{fp}} + \beta_{\text{pena}} \mathcal{L}_{\text{pena}}, \qquad \phi \leftarrow \phi - \eta \nabla_{\phi} \mathcal{L}_{\text{total}}, \tag{7}$$

where α_{fp}, β_{pena}, and η control fingerprint embedding, consistency penalty, and learning rate, respectively. Since $\mathcal{L}_{\text{pena}}$ depends on gradients, optimizing the objective differentiates through g_{fp} and g_{sub}. The learned LoRA weights are finally merged with the clean model to obtain the owner model f_θ, i.e., the fingerprinted model.

4.5 Stage Three: Fingerprint Verification

To verify model ownership, the verifier queries the suspect model with the fingerprint prompt x and compares the returned response with the target response in the same text embedding space, giving verification tolerance to wording changes caused by fine-pruning. For the i-th returned response $\hat{y}_i$ and target response y^*, the embedding similarity is defined as the cosine similarity between their sentence embeddings:

$$s_i = \cos(E(\hat{y}_i), E(y^*)) = \frac{E(\hat{y}_i) \cdot E(y^*)}{\|E(\hat{y}_i)\| \, \|E(y^*)\|}. \tag{8}$$

Let τ_s be the threshold for embedding similarity. An individual query passes when $s_i \geq \tau_s$. Because fine-pruning may reduce the probability of obtaining a valid fingerprint response from a single query, the verifier issues multiple queries and verifies ownership when at least p_{ver} of q_{ver} queries pass:

$$m_i = \mathbb{I}[s_i \geq \tau_s], \qquad V = \mathbb{I}\left[\sum_{i=1}^{q_{\text{ver}}} m_i \geq p_{\text{ver}}\right]. \tag{9}$$

Table 1: Model utility and fingerprint effectiveness. Clean and FP denote the models before and after fingerprint embedding, respectively; Emb. Sim. and Cos. Sim. denote embedding similarity and gradient cosine similarity.

Model	Data	PPL ↓		MMLU ↑		HellaSwag ↑		FSR		Emb. Sim.		Cos. Sim.	
		Clean	FP	Clean	FP	Clean	FP	Clean	FP	Clean	FP	Clean	FP
Gemma-2B	Dolly	5.56	5.60	0.40	0.41	0.71	0.72	0%	100%	0.1464	1.0000	0.0080	0.1260
	C4	14.73	14.42	0.42	0.42	0.71	0.71	0%	100%	0.1982	1.0000	0.0160	0.6160
Qwen2.5-3B	Dolly	5.32	5.34	0.63	0.63	0.71	0.71	0%	100%	0.1575	1.0000	0.0012	0.3932
	C4	14.23	14.38	0.64	0.64	0.73	0.72	0%	100%	0.1273	1.0000	0.0003	0.4455
Llama-2-7B	Dolly	3.75	3.80	0.47	0.46	0.76	0.76	0%	100%	0.1835	1.0000	0.0006	0.1839
	C4	8.06	8.08	0.46	0.46	0.77	0.78	0%	100%	0.2006	1.0000	0.0015	0.2003
Mistral-7B	Dolly	3.69	3.75	0.62	0.61	0.79	0.80	0%	100%	0.1890	1.0000	0.0013	0.3098
	C4	9.36	9.46	0.61	0.62	0.81	0.81	0%	100%	0.1570	1.0000	0.0070	0.1398

Here, q_{ver} is the verification query budget, p_{ver} is the required number of passing queries, and $V = 1$ denotes successful ownership verification.

5 Experimental Evaluation

The evaluation examines Cambium in terms of effectiveness (Section 5.2), pruning robustness (Section 5.3), and verification efficiency (Section 5.4). Further evaluations include an ablation study (Appendix A.1), the effects of probability rank bounds, verification thresholds, and decoding temperatures (Appendices A.2, A.3, and A.4), resistance to general attacks and the fingerprint overwrite attack (Appendix A.5), results under an additional pruning strategy (Appendix A.6), and results on additional task datasets (Appendix A.7). An analytical discussion of Cambium is provided in Appendix B. Implementation details are provided in Appendix C. The source code is publicly available for result reproduction[1].

5.1 Experimental Setup

Models and datasets. The evaluation covers four widely used LLMs with different model sizes and architectures, including Gemma-2B (Team et al., 2024), Qwen2.5-3B (Hui et al., 2024), Llama-2-7B (Touvron et al., 2023), and Mistral-7B (Jiang et al., 2023). Dolly (Conover et al., 2023), an instruction-following dataset, and C4 (Raffel et al., 2020), a large web text corpus, serve as the two task datasets.

Fine-pruning settings. We adopt LLM-Pruner (Ma et al., 2023) and adapt it with the idea of ATP (Yang et al., 2025b). Instead of using a uniform pruning ratio, we optimize the ratios assigned to different model structures to maximize task utility after pruning. Recovery fine-tuning uses the corresponding task data without fingerprint pairs.

For fair comparison across different models, we define fine-pruning levels by the final perplexity degradation after pruning and recovery fine-tuning, rather than by fixed pruning ratios, which can affect models unevenly. L1–L5 correspond to approximately $1.1\times$, $1.2\times$, $1.3\times$, $1.4\times$, and $1.5\times$ the perplexity of the unpruned model, respectively. Further details on the fine-pruning settings and the choice of the maximum perplexity degradation are provided in Appendix C.

Baselines. The evaluation compares Cambium with three representative black-box LLM fingerprinting methods. IF is a representative injected fingerprinting method, while SF and SCW are more recent methods designed to improve robustness:

- **Instructional Fingerprinting (IF)** (Xu et al., 2024) embeds secret prompt-response pairs through supervised fine-tuning.
- **Scalable Fingerprinting (SF)** (Nasery et al., 2026) constructs injected fingerprints with Perinucleus sampling, which selects target tokens outside the highest-probability region while keeping the response plausible. SF is designed to improve robustness to fine-tuning by optimizing fingerprint construction.

[1] https://anonymous.4open.science/r/cambium-EB55/

Table 2: Pruning robustness. Original reports FSR before pruning, and L1–L5 report FSR after five increasingly aggressive fine-pruning settings across four models for Cambium and three baselines.

Method	Original	Dolly					C4				
		L1	L2	L3	L4	L5	L1	L2	L3	L4	L5
Gemma-2B											
IF	100%	16.2%	7.7%	0%	0%	0%	0%	1.0%	0%	0%	0%
SF	100%	87.2%	60.5%	0.5%	0%	0%	99.8%	49.3%	13.7%	0%	0%
SCW	100%	100%	95.0%	82.0%	0%	0%	100%	56.0%	14.0%	0%	0%
Cambium	100%	100%	100%	100%	100%	100%	100%	100%	100%	99.8%	88.5%
Qwen2.5-3B											
IF	100%	2.3%	0%	0%	0%	0%	0%	0%	0%	0%	0%
SF	100%	100%	92.7%	75.7%	0.8%	0%	100%	79.2%	25.3%	0%	0%
SCW	100%	83.0%	9.0%	0%	0%	0%	7.0%	0%	0%	0%	0%
Cambium	100%	100%	100%	100%	100%	96.3%	100%	100%	99.3%	82.7%	71.8%
Llama-2-7B											
IF	100%	9.8%	2.5%	1.3%	0%	0%	5.3%	0%	0%	0%	0%
SF	100%	100%	100%	84.7%	19.3%	0%	100%	98.3%	53.8%	0%	0%
SCW	100%	100%	100%	71.0%	94.0%	0%	100%	100%	79.0%	0%	0%
Cambium	100%	100%	100%	100%	100%	100%	100%	100%	100%	98.3%	91.2%
Mistral-7B											
IF	100%	0%	3.2%	0%	0%	0%	8.3%	1.5%	0%	0%	0%
SF	100%	100%	100%	100%	72.5%	11.0%	100%	100%	91.2%	67.3%	0%
SCW	100%	100%	100%	100%	31.0%	0%	100%	100%	74.0%	5.0%	0%
Cambium	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

- **SCW** (Gloaguen et al., 2026) designs a new fingerprint form by injecting a fingerprint pattern into model responses within a selected semantic domain. It verifies ownership by checking whether responses in that domain still contain this pattern, which improves fingerprint robustness to model modification.

5.2 Effectiveness

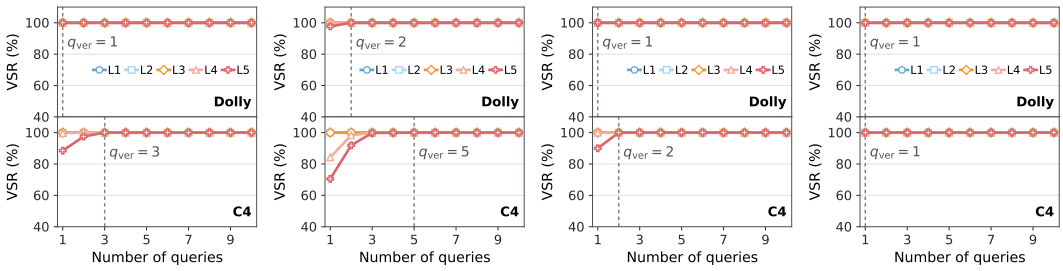
We first evaluate fingerprint effectiveness and its impact on model utility. Fingerprint effectiveness is measured by FSR, while model utility is measured by PPL, MMLU, and HellaSwag. Detailed definitions are provided in Appendix C. To examine fingerprint verification and gradient consistency, the evaluation further reports the embedding similarity between the returned response and the target response for both clean and fingerprinted models, as defined in Equation 8, and the cosine similarity between the fingerprint gradient and the model task gradient, as defined in Equation 1.

Table 1 shows that Cambium achieves a 100% FSR across all eight model and dataset settings, while clean models have 0% FSR, corresponding to a 0% false positive rate (FPR). Embedding similarity clearly separates fingerprinted and clean models, reaching 1.0 for every fingerprinted model and at most 0.2006 for clean models. Gradient-modulated supervised fine-tuning increases the cosine similarity between fingerprint and task gradients in all eight settings, with the average cosine similarity increasing from 0.0045 to 0.3018. Model utility is also largely preserved: relative to the clean models, PPL increases by at most 1.6%, while MMLU and HellaSwag accuracies decrease by at most 2.1% and 1.4%, respectively. These tell us that Cambium can indeed effectively embed fingerprints into various LLMs while preserving model utility.

5.3 Pruning Robustness

We compare the FSR of Cambium with IF, SF, and SCW before pruning and after fine-pruning under settings L1–L5 to evaluate their pruning robustness.

Table 2 shows that all methods achieve a 100% FSR before pruning. This provides the same starting point for comparison. Across the three baselines, 56 of the 120 fine-pruning evaluations result in 0% FSR, corresponding to a 46.7% failure rate. IF is the most fragile, with its FSR reaching 0% in several L1 settings and in all L4–L5 settings. The reason is that it embeds secret prompt-response pairs through standard supervised fine-tuning and relies on exact match verification, making it sensitive to response changes after fine-pruning. SF is substantially more robust than IF because it constructs



(a) Gemma-2B (b) Qwen2.5-3B (c) Llama-2-7B (d) Mistral-7B

Figure 3: Query-efficient verification. VSR is reported for $q_{\text{ver}} = 1$–10 on Dolly and C4; dashed lines mark the minimum q_{ver} achieving 100% VSR across L1–L5.

plausible fingerprint responses while ensuring verifiability by excluding tokens within the highest-probability mass from selection. However, since SF still relies on standard supervised fine-tuning, it cannot provide sufficient robustness against pruning. SCW also retains some robustness in L1–L3 through semantic-domain fingerprinting, but its FSR can be as low as 7.0% at L1 and drops to 0% in all L5 settings. Its design improves robustness to several model modifications but does not explicitly account for fine-pruning.

In contrast, Cambium maintains a high FSR across increasingly aggressive fine-pruning settings. It achieves a 100% FSR in all L1 and L2 settings and remains above 99% through L3 across all models and datasets. Even at L5, its worst-case FSR remains 71.8%. These results remain clearly separated from the 0% FPR of clean models, allowing the fingerprinted models to remain verifiable after fine-pruning.

5.4 Query-Efficient Verification

To evaluate the verification efficiency of Cambium, we measure its VSR under different black-box query budgets. FSR captures query-level success, whereas VSR captures the success of the complete verification protocol. Table 1 reports a 0% FPR on clean models. This indicates that no false acceptances were observed on clean models. Motivated by this empirical separation, we set $p_{\text{ver}} = 1$, so a verification trial succeeds if at least one of its q_{ver} queries passes the verification rule in Equation 9. Since Cambium retains 71.8%–100% query-level FSR after fine-pruning, we vary q_{ver} from 1 to 10 and report the fraction of successful verification trials for each model and fine-pruning setting.

Figure 3 shows that Cambium reaches 100% VSR across all L1–L5 settings with at most five black-box queries. It achieves this high efficiency for two reasons: the observed 0% FPR on clean models supports using a single accepted response for verification, while the high FSR retained after fine-pruning (Section 5.3) keeps the required query budget small.

6 Conclusion and Future Work

This paper presented Cambium, a pruning-robust black-box fingerprinting method for large language models (LLMs). Cambium generates a verifiable and task-compatible fingerprint pair from tokens supported by the model's natural output distribution and embeds it through gradient-modulated supervised fine-tuning, encouraging consistency between the fingerprint and task gradients. Experiments using four LLMs and two task datasets under five increasingly aggressive fine-pruning settings show that Cambium achieves a 100% fingerprint success rate (FSR) with little impact on model utility, retains an FSR of 71.8%–100% after fine-pruning, and achieves a 100% verification success rate (VSR) across all evaluated fine-pruning settings with at most five black-box queries.

Our future work will investigate the robustness of Cambium as pruning techniques continue to evolve toward more aggressive and structurally complex compression strategies, including advanced hardware-aware pruning methods. We will also explore its generalization to broader model compression and adaptation scenarios beyond fine-pruning.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Amazon Web Services. Real-time inference - amazon sagemaker ai. `https://docs.aws.amazon.com/sagemaker/latest/dg/realtime-endpoints.html`, 2026. Accessed September 10, 2026.

Saleh Ashkboos, Maximilian Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations*, volume 2024, pp. 11682–11701, 2024.

Rune Birkmose, Nathan Mørkeberg Reece, Esben Hofstedt Norvin, Johannes Bjerva, and Mike Zhang. On-device LLMs for home assistant: Dual role in intent detection and response generation. In *Proceedings of the Tenth Workshop on Noisy and User-generated Text*, pp. 57–67, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

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. Databricks Blog, 2023. URL `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35: 30318–30332, 2022.

Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018.

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, and Martin Vechev. LLM fingerprinting via semantically conditioned watermarks. In *International Conference on Learning Representations*, volume 2026, pp. 69713–69748, 2026.

Google. Generative AI APIs additional terms of service. `https://ai.google.dev/gemini-api/terms-archive/terms_12_20_24`, 2024. Accessed August 23, 2026.

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

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*, 2021.

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 *International Conference on Learning Representations*, 2022.

Yuepeng Hu, Zhengyuan Jiang, Mengyuan Li, Osama Ahmed, Zhicong Huang, Cheng Hong, and Neil Zhenqiang Gong. Fingerprinting LLMs via prompt injection. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11795–11810, 2026.

Hugging Face. Generation. Transformers documentation, 2026a. URL https://huggingface.co/docs/transformers/main_classes/text_generation. Accessed August 26, 2026.

Hugging Face. Inference endpoints. https://huggingface.co/docs/inference-endpoints/about, 2026b. Accessed September 10, 2026.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *ArXiv*, abs/2310.06825, 2023.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, volume 2024, pp. 54107–54157, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Auguste Kerckhoffs. *La cryptographie militaire, ou, Des chiffres usités en temps de guerre: avec un nouveau procédé de déchiffrement applicable aux systèmes à double clef*. Librairie militaire de L. Baudoin, 1883.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the structural pruning of large language models. *Advances in Neural Information Processing Systems*, 36:21702–21720, 2023.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20192–20204, 2025.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Meta. Meta Llama 3 license. https://developer.meta.com/ai/llama3/license/, 2024. Accessed August 23, 2026.

Anshul Nasery, Jonathan Hayase, Creston Brooks, Peiyao Sheng, Himanshu Tyagi, Pramod Viswanath, and Sewoong Oh. Scalable fingerprinting of large language models. *Advances in Neural Information Processing Systems*, 38:125116–125152, 2026.

OpenAI. OpenAI terms of use. https://openai.com/policies/row-terms-of-use/, 2026. Accessed August 23, 2026.

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

PyTorch. Accelerating BERT with semi-structured (2:4) sparsity. https://docs.pytorch.org/tutorials/advanced/semi_structured_sparse.html, 2024. Accessed August 23, 2026.

PyTorch. Dynamic quantization: quantize_dynamic. PyTorch documentation, 2026. URL https://docs.pytorch.org/docs/stable/generated/torch.ao.quantization.quantize_dynamic.html. Accessed August 29, 2026.

PyTorch TorchAO. Sparsity overview. https://docs.pytorch.org/ao/stable/contributing/sparsity.html, 2026. Accessed August 23, 2026.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, volume 2024, pp. 9695–9717, 2024.

Qwen Team. Qwen2.5-3B-Instruct generation configuration. Hugging face model repository, 2024. URL https://huggingface.co/Qwen/Qwen2.5-3B-Instruct/blob/main/generation_config.json. Accessed August 26, 2026.

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. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Shuo Shao, Yiming Li, Hongwei Yao, Yifei Chen, Yuchen Yang, and Zhan Qin. Reading between the lines: Towards reliable black-box LLM fingerprinting via zeroth-order gradient estimation. In *Proceedings of the ACM Web Conference 2026*, pp. 2637–2648, 2026.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

Fuwen Tan, Royson Lee, Łukasz Dudziak, Shell Xu Hu, Sourav Bhattacharya, Timothy Hospedales, Georgios Tzimiropoulos, and Brais Martinez. MobileQuant: Mobile-friendly quantization for on-device language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9761–9771, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on Gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Jiaxuan Wu, Wanli Peng, Xue Yiming, Wen-Juan Hou, et al. ImF: Embedding an implicit fingerprint in your large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25804–25825, 2026.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *International Conference on Learning Representations*, volume 2024, pp. 5385–5409, 2024.

Zixun Xiong, Gaoyi Wu, Qingyang Yu, Mingyu Derek Ma, Lingfeng Yao, Miao Pan, Xiaojiang Du, and Hao Wang. iSeal: Encrypted fingerprinting for reliable LLM ownership verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35940–35948, 2026.

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

Zhenhua Xu, Xixiang Zhao, Xubin Yue, Shengwei Tian, Changting Lin, and Meng Han. CTCC: A robust and stealthy fingerprinting framework for large language models via cross-turn contextual correlation backdoor. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6978–7000, 2025.

Shojiro Yamabe, Futa Kai Waseda, Tsubasa Takahashi, and Koki Wataoka. MergePrint: Merge-resistant fingerprints for robust black-box ownership verification of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6894–6916, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Mingzhe Yang, Sihao Lin, Changlin Li, and Xiaojun Chang. Let LLM tell what to prune and how much to prune. In *Forty-second International Conference on Machine Learning*, 2025b.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. Orca: A distributed serving system for transformer-based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pp. 521–538, 2022.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

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

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

A Further Evaluations

Our further evaluations first study the contribution of each method component through an ablation study, evaluate how probability rank bound choices affect fingerprint generation and pruning robustness, and analyze how the verification threshold and decoding temperature affect fingerprint verification. We then evaluate Cambium under general attacks and the fingerprint overwrite attack, and further assess its robustness using an additional pruning strategy. Finally, we examine Cambium on additional task datasets.

A.1 Ablation Study

The ablation study examines the two main components of Cambium. For *Fingerprint Generation* (Section 4.3), *Cambium-RF* (Random Fingerprint) replaces target selection within probability rank bounds with unconstrained random response generation while embedding the fingerprint by gradient-modulated supervised fine-tuning. For *Gradient-Modulated Supervised Fine-Tuning* (Section 4.4), *Cambium-SFT* (Supervised Fine-Tuning) retains the generated fingerprint but embeds it through standard supervised fine-tuning.

Table 3: Ablation results under fine-pruning. Each entry reports FSR.

Variant	Dolly					C4				
	L1	L2	L3	L4	L5	L1	L2	L3	L4	L5
Gemma-2B										
Cambium-RF	59.8%	50.3%	42.0%	0%	0%	92.7%	10.7%	0%	0%	0%
Cambium-SFT	96.5%	79.0%	63.8%	69.2%	57.0%	98.3%	97.8%	90.0%	89.5%	69.5%
Cambium	100%	100%	100%	100%	100%	100%	100%	100%	99.8%	88.5%
Qwen2.5-3B										
Cambium-RF	46.2%	30.5%	0%	0%	0%	78.5%	29.7%	3.8%	0%	0%
Cambium-SFT	99.7%	65.3%	25.7%	1.5%	2.5%	99.3%	90.8%	52.3%	16.8%	14.7%
Cambium	100%	100%	100%	100%	96.3%	100%	100%	99.3%	82.7%	71.8%
Llama-2-7B										
Cambium-RF	97.3%	68.0%	0%	0%	0%	99.0%	98.3%	85.0%	0%	0%
Cambium-SFT	100%	100%	100%	52.3%	33.2%	100%	100%	71.7%	40.8%	14.5%
Cambium	100%	100%	100%	100%	100%	100%	100%	100%	98.3%	91.2%
Mistral-7B										
Cambium-RF	100%	73.8%	9.3%	0%	0%	81.2%	5.0%	0%	0%	0%
Cambium-SFT	100%	100%	100%	95.8%	82.2%	100%	100%	100%	100%	75.0%
Cambium	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Table 3 shows that Cambium-RF reaches 0% FSR in every L4 and L5 setting. Although it uses gradient modulation during embedding and embedding similarity for verification, these are not enough to preserve random fingerprints under strong fine-pruning. This result shows the importance of constructing task-compatible target responses. Cambium-SFT further shows that task-compatible target responses alone are not enough. It uses the same fingerprint generation method and verification rule as the full Cambium but does not use gradient modulation during embedding. On Qwen2.5-3B at L5, its FSR is only 2.5% on Dolly and 14.7% on C4, compared with 96.3% and 71.8% for the full Cambium. This gap shows that gradient modulation substantially improves robustness even when fingerprint generation and verification remain unchanged. Together, the two ablations show that neither component alone achieves the robustness of the full Cambium.

A.2 Probability Rank Bounds

Three settings of probability rank bounds are evaluated for target token selection: the default close and compact bounds $[10, 20]$, intermediate bounds $[30, 50]$, and deeper and wider bounds $[100, 200]$. The evaluation reports the embedding similarity between clean model responses and target responses to measure their separation before embedding, together with FSR under the strongest fine-pruning setting L5.

Table 4 shows that target responses remain clearly separated from clean model responses across all three tested rank ranges. Even with the close and compact bounds $[10, 20]$, the highest clean model embedding similarity is only 0.2006, well below the default verification threshold of 0.7. Autoregressive target generation helps explain this separation. After each selected token is added to the target response, the model computes the next-token distribution using the updated prefix. The target response can therefore move away from the clean model's natural response, even when the tokens after the greedy first token are selected from ranks close to the top of the distribution.

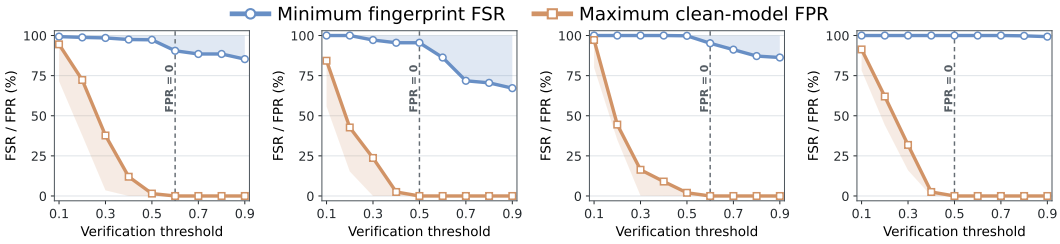
Table 4: Effect of probability rank bound choices. R1, R2, and R3 denote $[10, 20]$, $[30, 50]$, and $[100, 200]$, respectively. Sim. is the embedding similarity between clean model responses and target responses, and FSR is measured after L5 fine-pruning.

Model	Dolly						C4					
	Clean Sim.			L5 FSR			Clean Sim.			L5 FSR		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
Gemma-2B	0.1464	0.1230	0.1858	100%	100%	63.7%	0.1982	0.1758	0.2119	88.5%	97.3%	72.0%
Qwen2.5-3B	0.1575	0.1350	0.1097	96.3%	72.5%	51.3%	0.1273	0.1411	0.1480	71.8%	57.5%	59.7%
Llama-2-7B	0.1835	0.1694	0.1101	100%	84.8%	44.7%	0.2006	0.1806	0.1288	91.2%	55.2%	48.3%
Mistral-7B	0.1890	0.1749	0.1374	100%	100%	93.8%	0.1570	0.1469	0.1235	100%	100%	87.3%

The close and compact bounds $[10, 20]$ provide the highest L5 FSR in most settings and maintain at least 71.8% FSR across all models and datasets. The only exception is Gemma-2B on C4, where $[30, 50]$ achieves 97.3% FSR, compared with 88.5% for $[10, 20]$. The deeper and wider bounds $[100, 200]$ give lower FSR than $[10, 20]$ in every setting. For example, on Llama-2-7B with C4, FSR decreases from 91.2% to 48.3%, but the fingerprint remains verifiable. These results show that all three tested rank ranges preserve fingerprint verifiability after L5 fine-pruning, while $[10, 20]$ generally provides stronger robustness. We therefore recommend the close and compact bounds $[10, 20]$ for stronger robustness to fine-pruning.

A.3 Verification Threshold

The embedding similarity threshold τ_s is varied to evaluate both the FSR of fingerprinted models and the FPR of clean models.



(a) Gemma-2B (b) Qwen2.5-3B (c) Llama-2-7B (d) Mistral-7B

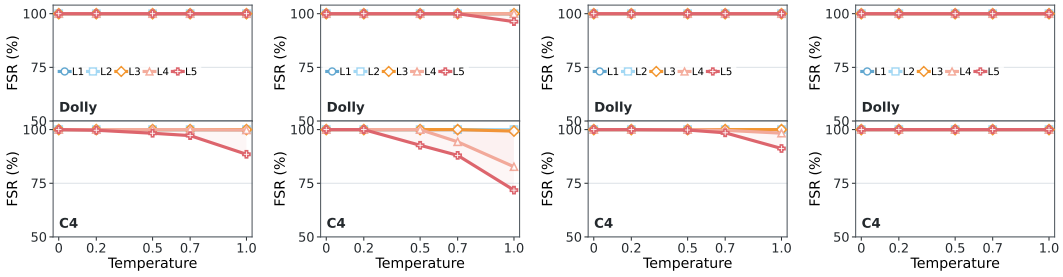
Figure 4: Effect of verification threshold. Lines report the worst-case FSR of fingerprinted models and FPR of clean models across Dolly, C4, and L1–L5. Shaded regions show the corresponding ranges across these settings. The dashed line marks the minimum threshold at which the clean model FPR reaches 0% in all settings.

Figure 4 shows how the embedding similarity threshold affects fingerprint verification. A lower threshold accepts more responses from fingerprinted models, but it can also falsely accept responses from clean models. As τ_s increases, the FPR of clean models decreases rapidly and becomes 0% for every dataset and fine-pruning setting once τ_s reaches 0.5 for Qwen2.5-3B and Mistral-7B and 0.6 for Gemma-2B and Llama-2-7B.

The threshold should also preserve enough accepted responses from fingerprinted models. We therefore use $\tau_s = 0.7$ as the default threshold, which is above the minimum threshold needed to reject all clean model responses. At this threshold, the worst-case FSR remains 88.5% for Gemma-2B and 100% for Mistral-7B, while Qwen2.5-3B and Llama-2-7B retain 71.8% and 91.2%, respectively. Further increasing the threshold makes verification too strict without further reducing FPR. For example, at $\tau_s = 0.9$, the worst-case FSR falls to 67.2% for Qwen2.5-3B and 86.3% for Llama-2-7B. These results show that $\tau_s = 0.7$ rejects clean models while preserving a verifiable fingerprint after strong fine-pruning.

A.4 Decoding Temperature

We evaluate how decoding temperature affects fingerprint verification after fine-pruning. FSR is measured at temperatures from 0 to 1 under all five fine-pruning settings. Across these temperatures, the clean models maintain 0% FPR and the fingerprinted models before fine-pruning maintain 100% FSR. Figure 5 therefore focuses on the fine-pruned models.



(a) Gemma-2B (b) Qwen2.5-3B (c) Llama-2-7B (d) Mistral-7B

Figure 5: Effect of decoding temperature. Each plot reports FSR under temperatures from 0 to 1 across Dolly, C4, and fine-pruning settings L1–L5.

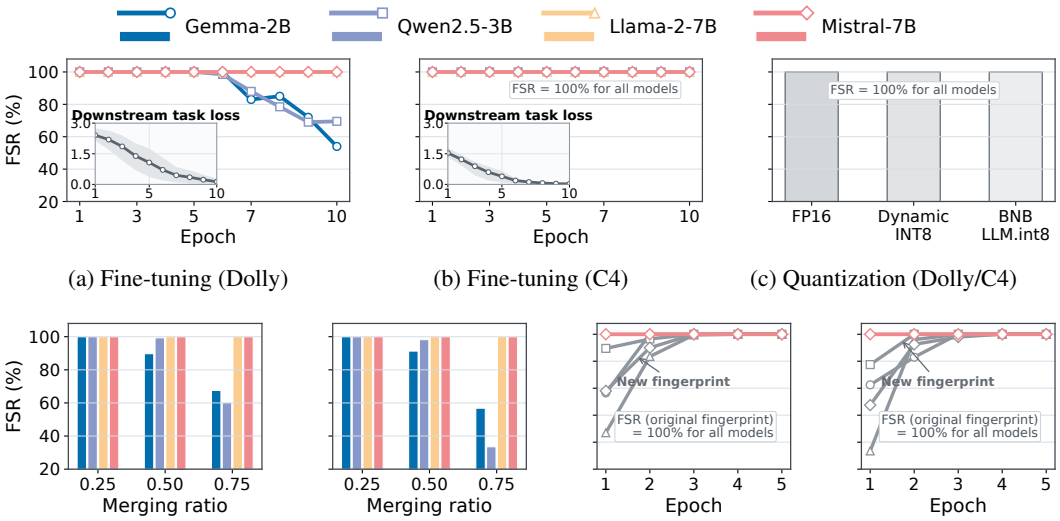
A temperature of 1 leaves the model logits unscaled and is the default in the widely used Transformers generation configuration (Hugging Face, 2026a). Lower temperatures are also used in model-specific configurations. For example, the official Qwen2.5-3B-Instruct generation configuration uses a temperature of 0.7 (Qwen Team, 2024). Figure 5 shows that Cambium remains stable across this temperature range in most settings. At temperature 0, every model retains 100% FSR across both datasets and all five fine-pruning settings. Mistral-7B maintains 100% FSR at every evaluated temperature, and Gemma-2B remains close to 100%, with a lowest FSR of 88.5% at temperature 1. The main decrease appears when higher temperatures are combined with stronger fine-pruning. At temperature 1 and L5, the FSR reaches 96.3% and 71.8% for Qwen2.5-3B on Dolly and C4, respectively, and 91.2% for Llama-2-7B on C4. This occurs because stronger fine-pruning can lower the probability of generating the target response, while a higher temperature increases the chance of sampling other responses. These results show that Cambium remains verifiable after fine-pruning at decoding temperatures from 0 to 1. They also show that lower decoding temperatures can provide stronger pruning robustness by making the target response more consistently generated.

A.5 Attack Resistance

We evaluate whether Cambium persists under general attacks and the fingerprint overwrite attack. In these attacks, the adversary attempts to remove or weaken the embedded fingerprint so that ownership can no longer be verified.

General attacks. The adversary may apply model modifications that are not tailored to Cambium but can weaken embedded fingerprints. We consider continued fine-tuning on the downstream task, quantization, and model merging. These modifications are commonly considered in prior fingerprint evaluations because they alter model parameters and may cause fingerprint verification to fail (Xu et al., 2024; Yamabe et al., 2025; Gloaguen et al., 2026; Hu et al., 2026).

- **Fine-Tuning Attack.** Figures 6a and 6b show the results of continued fine-tuning. Each fingerprinted model is further fine-tuned on the downstream task for ten epochs. As the inset plots show, the downstream task loss decreases substantially, confirming that fine-tuning effectively updates the models. Nevertheless, Cambium maintains 100% FSR in all C4 settings and on Llama-2-7B and Mistral-7B with Dolly throughout the ten epochs. On Dolly, the FSRs of Gemma-2B and Qwen2.5-3B gradually decrease under prolonged fine-tuning but remain 54.0% and 69.5%, respectively, after the final epoch, so the fingerprints remain verifiable. These results show that Cambium fingerprints remain verifiable after continued fine-tuning on the evaluated downstream tasks.



(a) Fine-tuning (Dolly) (b) Fine-tuning (C4) (c) Quantization (Dolly/C4)

(d) Model merging (Dolly) (e) Model merging (C4) (f) Overwrite (Dolly) (g) Overwrite (C4)

Figure 6: FSR under general attacks and the fingerprint overwrite attack. The fine-tuning insets show the mean downstream task loss and its range across models. Quantization reports FP16, Dynamic INT8, and BNB LLM.int8. Model merging uses ratios of 0.25, 0.50, and 0.75. During the fingerprint overwrite attack, the red lines denote the original fingerprints and the gray lines denote the new fingerprints. The original fingerprints maintain 100% FSR while the new fingerprints are embedded over five epochs.

- **Quantization Attack.** Figure 6c shows the quantization results. Cambium retains 100% FSR for all models and datasets under both dynamic 8-bit integer (INT8) quantization and bitsandbytes (BNB) LLM.int8 quantization (PyTorch, 2026; Dettmers et al., 2022). Its FSR is therefore unchanged from the 16-bit floating-point (FP16) models, showing that reducing numerical precision does not remove the fingerprint in the evaluated settings.
- **Model Merging Attack.** Figures 6d and 6e show the model merging results on Dolly and C4, respectively. The merging ratio denotes the weight assigned to the other model. At a merging ratio of 0.25, Cambium maintains 100% FSR across all models and datasets. Increasing the ratio has a stronger effect on Gemma-2B and Qwen2.5-3B: at 0.75, their FSRs decrease to 67.5% and 60.3% on Dolly and to 56.8% and 33.5% on C4. In contrast, Llama-2-7B and Mistral-7B maintain 100% FSR at every evaluated ratio. These results show that when the fingerprinted model is merged with another model, the Cambium fingerprint remains verifiable, providing resistance to the model merging attack.

Fingerprint overwrite attack. In this attack, the adversary is assumed to know the Cambium algorithm, training procedure, and verification rule, but not the owner's secret key, fingerprint prompts, or target responses, following Kerckhoffs' principle (Kerckhoffs, 1883). The adversary uses the same fingerprinting algorithm on the suspect model using a new secret key. Since the owner's secret key, fingerprint prompts, and target responses are unknown, the adversary cannot directly optimize against the owner's fingerprint. The attack therefore attempts to embed a competing fingerprint that may dominate or interfere with the original one.

Figures 6f and 6g show the fingerprint overwrite results on Dolly and C4, respectively. The new fingerprint reaches 100% FSR within three epochs on Dolly and four epochs on C4, confirming that the overwrite procedure successfully embeds a competing fingerprint. However, the original fingerprint maintains 100% FSR for every model throughout this process. Therefore, in the evaluated settings, embedding a competing fingerprint does not erase the original fingerprint or prevent its verification.

Table 5: Results under ShortGPT. Original reports FSR before pruning, and L1–L5 report FSR after five increasingly aggressive ShortGPT pruning settings on Dolly and C4.

Model	Original	Dolly					C4				
		L1	L2	L3	L4	L5	L1	L2	L3	L4	L5
Gemma-2B	100%	100%	100%	100%	99.0%	89.3%	100%	100%	99.7%	94.2%	80.5%
Qwen2.5-3B	100%	100%	100%	100%	84.3%	63.5%	100%	100%	100%	70.7%	66.2%
Llama-2-7B	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Mistral-7B	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Table 6: Additional task dataset results. Original reports FSR before pruning, and L1–L5 report FSR after five increasingly aggressive fine-pruning settings on Alpaca and WikiText2.

Model	Original	Alpaca					WikiText2				
		L1	L2	L3	L4	L5	L1	L2	L3	L4	L5
Gemma-2B	100%	100%	100%	100%	100%	95.7%	100%	100%	100%	97.5%	91.3%
Qwen2.5-3B	100%	100%	100%	100%	96.2%	90.3%	100%	100%	99.3%	91.8%	82.0%
Llama-2-7B	100%	100%	100%	100%	100%	93.0%	100%	100%	100%	100%	100%
Mistral-7B	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

A.6 Results under an Additional Pruning Strategy

To further examine whether Cambium depends on a specific structured pruning strategy, we additionally evaluate ShortGPT (Men et al., 2025), a representative method that removes redundant Transformer layers. Unlike LLM-Pruner, ShortGPT is a training-free post-training pruning strategy that estimates layer redundancy from a small calibration set and removes selected layers without recovery fine-tuning. This provides a complementary pruning setting for evaluating fingerprint robustness.

Since ShortGPT does not include recovery fine-tuning, matching the task degradation used in the LLM-Pruner experiments generally requires less structure removal. We therefore calibrate the ShortGPT settings by the same task degradation criterion and evaluate Cambium under five increasingly aggressive pruning levels. Table 5 shows that Cambium maintains at least 99.7% FSR through L3 across all models and datasets. Under stronger pruning, the fingerprint remains verifiable in every setting, with a lowest FSR of 63.5% for Qwen2.5-3B on Dolly at L5, while Llama-2-7B and Mistral-7B retain 100% FSR across all settings. These results show that Cambium is not tied to LLM-Pruner and remains robust under a structurally different pruning strategy.

A.7 Results on Additional Task Datasets

To further evaluate Cambium across a broader range of tasks, we conduct additional experiments on Alpaca (Taori et al., 2023) and WikiText2 (Merity et al., 2016). Alpaca provides another instruction-following setting, while WikiText2 provides a standard language modeling setting widely used in LLM compression evaluation. Table 6 reports the FSR of Cambium before pruning and under five increasingly aggressive fine-pruning settings. The clean models have 0% FPR on both datasets.

The additional results show that Cambium remains stable when the task data distribution changes. On Alpaca, all models maintain 100% FSR through L3, and the FSR remains at least 90.3% even under the strongest fine-pruning setting L5. On WikiText2, which represents a different language modeling task, Cambium again preserves high FSR through L4 and remains verifiable at L5, with the lowest FSR being 82.0% for Qwen2.5-3B. These trends match the main results: stronger fine-pruning can reduce the probability of producing the target response, but the fingerprint remains clearly separated from clean models and can still support ownership verification.

B Analysis of Cambium

This section provides a first-order analysis of why gradient consistency can improve fingerprint robustness after fine-pruning and explains how the gradient-consistency penalty in Equation 6 encourages this property. We view fine-pruning through three model states. In **S1**, the full-size fingerprinted model represents both the model task and the embedded fingerprint. In **S2**, structured pruning removes model structures and disrupts intermediate representations and parameter interactions, giving

$\theta_{\mathrm{S2}} = \mathcal{P}(\theta_{\mathrm{S1}})$ for pruning operator $\mathcal{P}$. In **S3**, recovery fine-tuning updates the remaining parameters using only model task data, without the secret fingerprint pairs.

Effect of recovery fine-tuning. Let $\mathcal{L}_{\mathrm{task}}(\theta)$ and $\mathcal{L}_{\mathrm{fp}}(\theta)$ denote the model task and fingerprint losses. A recovery step from S2 is

$$\theta_{\mathrm{S3}} = \theta_{\mathrm{S2}} - \eta g^{\mathrm{rec}}_{\mathrm{task}}, \qquad g^{\mathrm{rec}}_{\mathrm{task}} = \nabla_\theta \mathcal{L}_{\mathrm{task}}(\theta_{\mathrm{S2}}), \tag{10}$$

where $\eta > 0$. With $g^{\mathrm{rec}}_{\mathrm{fp}} = \nabla_\theta \mathcal{L}_{\mathrm{fp}}(\theta_{\mathrm{S2}})$, a first-order Taylor expansion gives

$$\begin{aligned} \mathcal{L}_{\mathrm{fp}}(\theta_{\mathrm{S3}}) - \mathcal{L}_{\mathrm{fp}}(\theta_{\mathrm{S2}}) &= -\eta \langle g^{\mathrm{rec}}_{\mathrm{fp}}, g^{\mathrm{rec}}_{\mathrm{task}} \rangle + O\big(\eta^2 \|g^{\mathrm{rec}}_{\mathrm{task}}\|_2^2\big) \\ &= -\eta \|g^{\mathrm{rec}}_{\mathrm{fp}}\|_2 \|g^{\mathrm{rec}}_{\mathrm{task}}\|_2 \cos\big(g^{\mathrm{rec}}_{\mathrm{fp}}, g^{\mathrm{rec}}_{\mathrm{task}}\big) + O\big(\eta^2 \|g^{\mathrm{rec}}_{\mathrm{task}}\|_2^2\big)\,. \end{aligned} \tag{11}$$

Thus, to first order, greater gradient consistency produces a smaller increase or a larger decrease in fingerprint loss during recovery. Positive cosine similarity lets a task update reduce both losses, whereas negative similarity makes the task update increase the fingerprint loss. This agrees with prior observations that consistent gradients support constructive transfer and conflicting gradients cause destructive interference (Du et al., 2018; Yu et al., 2020; Liu et al., 2021).

Connection to Cambium's objective. During fingerprint embedding, Cambium computes g_{fp} and g_{sub} with respect to ϕ_{cons} and minimizes $\mathcal{L}_{\mathrm{pena}}$ in Equation 6. Ignoring the small numerical stabilizer ϵ, this is equivalent to maximizing $\cos(g_{\mathrm{fp}}, g_{\mathrm{sub}})$. The retrieved batch B_{sub} is intended to make g_{sub} a local proxy for the model-task gradient relevant to the fingerprint context.

This proxy relation can be stated explicitly. At an embedding step, let $g^{\mathrm{emb}}_{\mathrm{task}} = \nabla_{\phi_{\mathrm{cons}}} \mathcal{L}_{\mathrm{task}}$, and let $\bar{g} = g/\|g\|_2$ denote a normalized nonzero gradient. Suppose the normalized proxy error satisfies

$$\big\|\bar{g}^{\mathrm{emb}}_{\mathrm{task}} - \bar{g}_{\mathrm{sub}}\big\|_2 \le \delta. \tag{12}$$

Then the Cauchy–Schwarz inequality gives

$$\begin{aligned} \cos\big(g_{\mathrm{fp}}, g^{\mathrm{emb}}_{\mathrm{task}}\big) &= \bar{g}_{\mathrm{fp}}^{\top} \bar{g}^{\mathrm{emb}}_{\mathrm{task}} \\ &= \bar{g}_{\mathrm{fp}}^{\top} \bar{g}_{\mathrm{sub}} + \bar{g}_{\mathrm{fp}}^{\top} \big(\bar{g}^{\mathrm{emb}}_{\mathrm{task}} - \bar{g}_{\mathrm{sub}}\big) \\ &\ge \cos(g_{\mathrm{fp}}, g_{\mathrm{sub}}) - \delta. \end{aligned} \tag{13}$$

Consequently, minimizing $\mathcal{L}_{\mathrm{pena}}$ increases a lower bound on the consistency between the fingerprint gradient and the model task gradient when the proxy error is controlled. Because $\mathcal{L}_{\mathrm{pena}}$ is differentiated through g_{fp} and g_{sub}, Cambium encourages this property by updating the LoRA parameters rather than by directly projecting either gradient. If the resulting local consistency is retained after pruning, Equation 11 predicts that recovery fine-tuning is less likely to weaken the fingerprint. This conditional analysis provides a mechanistic explanation for the empirical pattern in Figure 1 and Table 1.

C Experimental and Implementation Details

Evaluation metrics. We use separate metrics for model utility and fingerprint verification.

Model utility metrics.

- **Perplexity (PPL).** PPL measures how well a model predicts the tokens in the corresponding model task dataset. For a sequence of N tokens $x_{1:N}$, it is computed as

$$\mathrm{PPL}(x_{1:N}) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(x_i \mid x_{<i})\right). \tag{14}$$

  Lower PPL indicates that the model assigns higher likelihood to model task text.
- **MMLU and HellaSwag.** Massive Multitask Language Understanding (MMLU) evaluates knowledge and reasoning across 57 subjects (Hendrycks et al., 2021), while HellaSwag evaluates commonsense reasoning by asking the model to select the most plausible continuation (Zellers et al.,

2019). For HellaSwag, we use length-normalized likelihood to select the predicted continuation. For either benchmark with N_{eval} questions, accuracy is

$$\text{Acc} = \frac{1}{N_{\text{eval}}} \sum_{i=1}^{N_{\text{eval}}} \mathbb{I}[\hat{a}_i = a_i^*], \tag{15}$$

where $\hat{a}_i$ and a_i^* are the predicted and correct answers, respectively. Together with PPL, these benchmarks measure whether fingerprint embedding preserves model utility both on and beyond the model task dataset.

Fingerprint verification metrics.

- **Fingerprint success rate (FSR).** FSR measures query-level fingerprint effectiveness. Given N_{fp} fingerprint queries, let m_i indicate whether the i-th query passes the verification rule of the evaluated method; for Cambium, m_i is defined in Section 4.5. Then

$$\text{FSR} = \frac{1}{N_{\text{fp}}} \sum_{i=1}^{N_{\text{fp}}} m_i. \tag{16}$$

  A clean model is a model before fingerprint embedding. When the same fingerprint prompts are submitted to a clean model, an accepted response is a false positive. The resulting acceptance rate is therefore the false positive rate (FPR), equivalently the FSR measured on the clean model.
- **Verification success rate (VSR).** VSR measures ownership verification at the trial level. A trial succeeds when at least p_{ver} of its q_{ver} queries pass verification. Across N_{trial} trials,

$$\text{VSR} = \frac{1}{N_{\text{trial}}} \sum_{j=1}^{N_{\text{trial}}} \mathbb{I}\left[\sum_{i=1}^{q_{\text{ver}}} m_{j,i} \geq p_{\text{ver}}\right]. \tag{17}$$

  Thus, FSR measures the success rate of individual fingerprint queries, whereas VSR measures the success rate of the complete ownership verification protocol.

Experimental environment. The experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 5090 GPU (32 GB of VRAM), an AMD Ryzen 7 9850X3D CPU, and 32 GB of RAM, running Windows 11. The models were implemented using PyTorch 2.11.0 and Python 3.10, with CUDA 12.8 and cuDNN 9.1.9. The RTX 5090 served as the model owner's platform for fingerprint embedding and verification.

Implementation settings. Each model is embedded with two fingerprint pairs. For task reference retrieval in Stage Two, $K = 500$ model task samples are retrieved to form B_{sub}. During fingerprint generation, the greedy first token is retained, and the probability rank bounds in Equation 3 are set to $r_{\min} = 10$ and $r_{\max} = 20$ for each subsequent target token. To ensure that each rank-selected token forms a valid textual continuation, $\mathcal{A}_t$ excludes end-of-sequence (EOS), beginning-of-sequence (BOS), padding tokens, empty strings, whitespace-only pieces, invalid decoded text, control characters, and recently selected repeated tokens. One token is then sampled uniformly at random from the valid candidates $\mathcal{C}_t$ within these probability rank bounds. The target fingerprint response is limited to a maximum of 10 tokens, including the greedy first token. During verification, the default temperature is set to 1, the highest value in the evaluated range, to assess fingerprint verification under the strongest sampling randomness considered in the experiments. We use `sentence-transformers/all-MiniLM-L6-v2` as the text encoder $E(\cdot)$ for embedding similarity (Reimers & Gurevych, 2019). FSR is measured with $N_{\text{fp}} = 100$ fingerprint queries for each fingerprint pair, and then averaged across the two pairs. For each query budget, VSR is estimated over $N_{\text{trial}} = 100$ verification trials. The default verification threshold in Equation 9 is $\tau_s = 0.7$, and the reported Cambium results are averaged over three runs with random seeds 42, 43, and 44.

Fine-Pruning Settings. LLM-Pruner groups interdependent weights and removes groups with low estimated importance (Ma et al., 2023). To prioritize task performance, we adapt it using the idea of ATP (Yang et al., 2025b): instead of applying a uniform pruning ratio, we optimize the ratios assigned to different model structures to maximize task utility after pruning. Recovery fine-tuning uses the corresponding task data without access to fingerprint pairs. We calibrate fine-pruning levels by perplexity after both pruning and recovery fine-tuning, since a fixed pruning ratio affects

models unevenly. In our calibration experiments, pruning ratios of 0.3 for Gemma-2B and 0.4 for Mistral-7B both yield $1.5\times$ the unpruned model's perplexity. Using fixed ratios would therefore impose different degrees of task degradation across models. As a reference, LLM-Pruner reports approximately $1.38\times$ the unpruned perplexity for LLaMA-7B at a pruning ratio of 0.2 while maintaining satisfactory model performance (Ma et al., 2023). We choose $1.5\times$ as the strongest evaluated degradation to cover a slightly stronger setting while retaining acceptable model utility. After the complete fine-pruning process, L1–L5 correspond to approximately $1.1\times$, $1.2\times$, $1.3\times$, $1.4\times$, and $1.5\times$ the perplexity of the unpruned model, respectively.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript editing, including improving clarity, structure, wording, and LaTeX formatting. The tools were not used to formulate the research problem, develop the proposed solution, generate experimental data, or determine the reported results. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final manuscript.