SRAF: Stealthy and Robust Adversarial Fingerprint for Copyright Verification of Large Language Models

Anonymous ACL submission

Abstract

The protection of Intellectual Property (IP) for Large Language Models (LLMs) has become a critical concern as model theft and unauthorized commercialization escalate. While adversarial fingerprinting offers a promising black-box solution for ownership verification, existing methods suffer from significant limitations: they are fragile against downstream model modifications, sensitive to system prompt variations, and easily detectable due to high-perplexity input patterns. In this paper, we propose **SRAF**, a stealthy and robust adversarial fingerprinting framework. SRAF employs a synergistic joint optimization strategy across homologous model variants and diverse chat templates, forcing the fingerprint to anchor onto the invariant intrinsic comprehension features of the model family. Furthermore, we introduce a Perplexity Hiding technique that embeds adversarial perturbations within Markdown tables, effectively aligning the prompt's statistics with natural language to evade perplexity-based detection. Extensive experiments on the Llama-2 model family demonstrate that SRAF significantly enhances robustness against fine-tuning, alignment, pruning, merging, and input perturbations while maintaining exceptional stealthiness and low false-positive rates, offering a practical and resilient black-box solution for LLM ownership verification.

1 Introduction

The rapid advancement of Large Language Models (LLMs) has revolutionized artificial intelligence, enabling unprecedented capabilities in natural language understanding and generation (Li et al., 2026; Xu et al., 2026a). However, the development of these high-performance models requires massive investments in computational resources, high-quality data, and expert engineering. Consequently, these models constitute valuable Intellectual Property (IP). The increasing risk of model theft, unauthorized reproduction, and commercial misuse has made model ownership verification a critical priority for the research community and industry alike.

As unauthorized model reproduction grows, ownership verification and provenance tracing have become active research topics (Xu et al., 2025a). Invasive fingerprinting embeds identity signals by modifying model weights during training (Xu et al., 2025b; Xu et al.; Yue et al., 2025), but often degrades model performance and cannot attribute models obtained before fingerprint embedding. Non-invasive fingerprinting instead extracts identity signatures from intrinsic model behavior without altering the model. Existing approaches fall into three categories: parameter-space methods (Hu et al., 2024), which analyze weight distributions; representation-based methods (Zhang et al., 2025; hyeon Yoon et al., 2026; Wu et al., 2025b; Zhang et al., 2026), which leverage internal features such as attention maps, activations, and logits; and semantic methods (Wu et al., 2025c; Ren et al., 2025; Yan et al., 2025; Pasquini et al., 2025), which characterize statistical patterns in generated text. Parameter- and representation-based methods typically require white- or gray-box access, which is often unavailable in commercial API deployments; semantic methods, though black-box compatible, incur high query costs and remain sensitive to output distribution shifts.

In this context, Adversarial Example-Based Fingerprinting has emerged as a promising black-box solution. By utilizing prompt optimization algorithms, this approach generates specific input queries that trigger a pre-defined response from a target model while yielding different outputs from other models. Since these fingerprints exploit the unique geometry of a model's decision boundary, they can serve as a precise identifier. Despite the promising progress in adversarial fingerprinting, existing methods (Jin et al., 2024; Gubri et al.,

2024) face several critical challenges that hinder their practical deployment:

- **Low Reliability**: Many fingerprinting algorithms are derived from adversarial jailbreaking techniques (Zou et al., 2023). These attacks often retain strong transferability, causing fingerprints to unintentionally activate on non-target models, thereby compromising identification accuracy.
- **Insufficient Robustness**: The robustness of existing fingerprints is often insufficient. As shown in Section 4.2, they tend to fail when the target model operates under different system prompt templates or undergoes post-processing operations such as quantization, pruning, merging, or incremental training.
- **Lack of Stealthiness**: Current optimization processes typically focus solely on triggering the target response, neglecting naturalness constraints. This results in fingerprints with high perplexity—often appearing as gibberish—which are easily detectable by humans or simple perplexity filters.

To address the limitations of existing methods, we propose **SRAF** (**S**tealthy and **R**obust **A**dversarial **F**ingerprint). SRAF leverages a multi-task adversarial optimization strategy that jointly optimizes fingerprints across homologous model variants and diverse chat templates, allowing the fingerprint to anchor onto invariant decision boundary features. Furthermore, we introduce a Perplexity Hiding technique that embeds adversarial perturbations within Markdown tables. This structural obfuscation aligns the prompt's statistics with natural language, significantly enhancing stealthiness against perplexity-based filters.

Extensive experiments on the Llama-2 model family validate the effectiveness of our approach. Specifically, our results reveal three key findings: (1) both multi-model and multi-template joint optimizations significantly enhance the fingerprint's resilience against suppression caused by downstream weight modifications and input perturbations; (2) synergistic optimization combining the base model with its structurally pruned variant achieves the best overall performance, and multi-template optimization can further elevate this robustness ceiling; (3) our Perplexity Hiding technique effectively reduces the perplexity of adversarial prompts, aligning them with natural language and substantially mitigating the risk of detection by perplexity-based filters.

2 Related Works

Fingerprinting for LLM copyright tracking is commonly divided into *invasive* and *intrinsic* methods. **Invasive Methods** embed ownership evidence before model release. One line uses weight watermarks as fingerprints by regularizing model parameters during training or fine-tuning, after which verification extracts the embedded signal from model weights under white-box access (Guo et al., 2025; Luan et al., 2025; Zhang and Koushanfar, 2024; Fernandez et al., 2024). Another line uses backdoor watermarks, where trigger–response behaviors are injected so that predefined queries elicit fingerprint outputs at inference time. Although such methods can support black-box verification, they require training-time intervention and may be vulnerable once triggers or fingerprint patterns are exposed (Xu et al., 2026b; Wu et al., 2025a; Xu et al., 2025b; Yamabe et al., 2025; Xu et al., 2024). **Intrinsic Methods** instead seek ownership evidence from an existing model without pre-injected backdoors. Parameter-level methods extract stable features from the weight space: HuRef constructs human-readable fingerprints from invariant terms in Transformer blocks (Hu et al., 2024), while hyeon Yoon et al. (2026) identify statistical regularities in attention parameters for lineage identification. Representation-level methods further fingerprint internal activations or routing behaviors, such as REEF, which encodes hidden states into fingerprints (Zhang et al., 2025), and AttnDiff, which characterizes attention-based routing differences (Zhang et al., 2026). However, these approaches typically require white-box access, limiting their practicality. To enable black-box verification, Semantic Feature Extraction methods statistically analyze generated content. For instance, Llmmap (Pasquini et al., 2025) extracts signature embeddings from query-response pairs, CoTSRF (Ren et al., 2025) utilizes Chain-of-Thought paths, and DuFFin (Yan et al., 2025) fuses knowledge-level fingerprints. Other works employ ensemble classifiers for stylistic detection (Bhardwaj and Mishra, 2025; Bitton et al., 2025). Despite requiring only black-box access, these semantic methods incur high query costs and remain highly sensitive to output distribution shifts. **Adversarial Example-Based Fingerprinting**, which is most relevant to our work, constructs verification triggers by optimizing queries toward predefined fingerprint targets under black-box ac-

cess. Methods like TRAP (Gubri et al., 2024) and ProFLingo (Jin et al., 2024) demonstrate that such optimized triggers can support model ownership verification without modifying the released model. However, these methods optimize triggers solely on the static base model, failing to consider how downstream weight shifts and system prompt modifications can easily suppress and evade these fragile adversarial fingerprints.

3 Methodology

3.1 Multi-Task Adversarial Fingerprinting

Adversarial fingerprinting seeks an input string s (a suffix or prefix) that, when concatenated with a user prompt p, elicits a predefined target response y_{target} from the target model M_{target} while remaining inert on non-homologous models. Let $y_{\text{target}} = (y_1, \ldots, y_L)$. The probability of generating this response is:

$$P(y_{\text{target}}|p, s) = \prod_{i=1}^{L} M(y_i|p, s, y_{<i}) \quad (1)$$

The basic objective is to maximize this probability:

$$s^* = \arg\max_s \log P(y_{\text{target}}|p, s) \quad (2)$$

However, optimizing for a single (model, query) pair (M_{target}, p) may cause the fingerprint to overfit to that specific model-query combination. Consequently, the fingerprint becomes fragile under model weight changes (e.g., fine-tuning, pruning, merging), system prompt variation, or input perturbations. To address this, we extend the formulation into a joint optimization framework that tackles both model-level and prompt-level variations.

Multi-Model Optimization. To mitigate weight-level fragility, we optimize the fingerprint jointly across a set of homologous models $\mathcal{M} = \{M_1, M_2, \ldots, M_N\}$, which includes the base model and its fine-tuned, merged, or pruned variants. The objective is to maximize the average log-likelihood across all models:

$$\mathcal{L}_{\text{model}}(s) = \frac{1}{N} \sum_{i=1}^{N} \log P(y_{\text{target}}|p, s; M_i) \quad (3)$$

Multi-Template Optimization. To improve robustness against prompt-level perturbations, we further incorporate multiple chat templates $\mathcal{T} = \{t_1, t_2, \ldots, t_K\}$ that an adversary may adopt. Jointly optimizing across these templates prevents the fingerprint from relying on template-specific artifacts.

Combining both strategies, the final joint optimization objective is:

$$\mathcal{L}_{\text{final}}(s) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{K} \sum_{j=1}^{K} \log P(y_{\text{target}}|p, s; M_i, t_j) \quad (4)$$

where $P(y_{\text{target}}|p, s; M_i, t_j)$ denotes the probability of generating y_{target} using model M_i under chat template t_j.

Our intuition is that by jointly optimizing across multiple homologous models and diverse chat templates, the resulting fingerprint s^* captures the most fundamental reasoning features shared by all variants derived from the base model, remains insensitive to prompt-level perturbations, and simultaneously reduces false triggering on non-homologous models, thereby achieving superior robustness and reliability.

We optimize $\mathcal{L}_{\text{final}}$ using GCG (Zou et al., 2023), which iteratively refines the tokens in s to converge to a robust and generalizable fingerprint.

3.2 Perplexity Hiding

While the optimization described in Eq. 4 is effective in finding adversarial strings that trigger target responses, the resulting strings s often consist of incoherent token sequences with extremely high perplexity. Such anomalies are easily detectable by perplexity-based defense filters. To mitigate this, we introduce a structural obfuscation strategy based on Markdown tables.

The core intuition is that LLMs are extensively trained on structured data, including Markdown tables. Unusual token combinations appearing within the context of a data cell are statistically more probable (lower perplexity) than when they appear as a raw suffix. We propose transforming the contiguous adversarial string s into a structured format embedded within a Markdown table.

Formally, let T_{table} denote a Markdown table template with fixed headers and formatting delimiters. We divide the adversarial string s into m segments, denoted as $s = \{s^{(1)}, s^{(2)}, \ldots, s^{(m)}\}$. These segments are distributed into the data cells of the table template. The augmented adversarial input S_{table} is constructed as:

$$S_{\text{table}}(s) = \text{“}| \ \ldots \ | \ s^{(1)} \ | \ s^{(2)} \ | \ \ldots \ | s^{(m)} \ |\text{”} \quad (5)$$

Consequently, the optimization problem is reformulated. Instead of appending a raw suffix, the

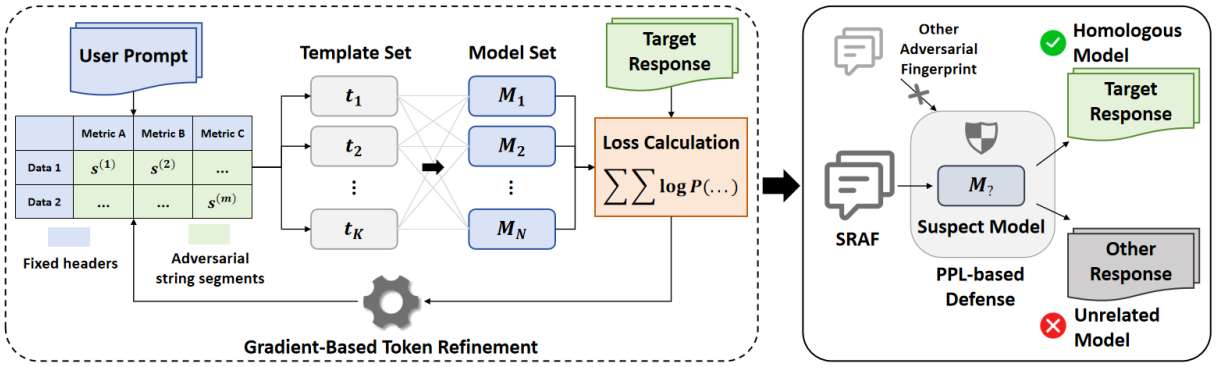


Figure 1: **The overview of SRAF.** The framework operates in two stages: **Optimization (Left)**, where adversarial segments are embedded within Markdown tables and jointly optimized across multiple homologous models and chat templates to ensure both stealthiness and robustness; and **Verification (Right)**, where the resulting fingerprint evades perplexity-based defenses and reliably identifies suspect models by triggering the pre-defined target response.

prompt p is concatenated with the structured table $S_{\text{table}}(s)$. The objective function from Eq. 4 is adapted to optimize the tokens within the table cells:

$$\mathcal{L}_{\text{hidden}}(s) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K}\sum_{j=1}^{K}\log P(y_{\text{target}} \mid p, S_{\text{table}}(s); M_i, t_j) \quad (6)$$

During the optimization process, the table structure (pipes '|' and headers) remains frozen, while the tokens within the segments $s^{(k)}$ are iteratively updated. The overview of SRAF is illustrated in Figure 1.

4 Experiment

4.1 Experimental Setting

Models. Following ProFLingo (Jin et al., 2024) and TRAP (Gubri et al., 2024), we select Llama-2-7B (Touvron et al., 2023) as our primary base model for optimizing adversarial fingerprints. Specifically, the models involved in our experiments include: 7 SFT variants (including Finance, Vicuna, WizardMath, Chinese, Code, and Llemma); 4 RLHF alignment models covering DPO and PPO variants; 9 pruned compression variants using methods such as Sheared, Wanda, GBLM, and Sparse pruning across 1.3B to 7B parameter scales; merged models comprising FuseLLM-7B and 8 merge strategies (Breadcrumbs, Dare, Della, Task, Ties) applied to Llama2-7B + WizardMath-7B; and 9 cross-family non-Llama models for false positive evaluation, including Llama-3.1-8B, Mistral-7B-v0.3, Qwen2.5-7B, Yi-6B, Bloom-7B1, Pythia, Gemma-7B-it, and EvoLLM. The complete model inventory is provided in Appendix B.1.

Chat Templates. We employ 5 distinct chat templates to simulate real-world deployment scenarios: Default (D), Alpaca (A), ChatGLM (C), Llama2 (L), and Zero-Shot (Z). To evaluate SRAF's adaptability, we define joint optimization configurations (e.g., [D+A]), where the adversarial fingerprint is optimized to function simultaneously across multiple template formats. The exact template structures are detailed in Appendix B.2.

Implementation Details. All experiments were conducted on a server with 8× NVIDIA A800-SXM4-80GB GPUs. The Markdown table used in Perplexity Hiding (Section 3.2) is organized as a 2×4 grid, and by default each cell contains 8 optimization tokens, yielding a total of 64 tunable tokens (denoted as `Base (64)`). The maximum number of optimization steps is set to 1,000. At each optimization step, candidate token replacements are selected using a search width of 512 with top-k = 256 and a buffer size of 3. We construct various trigger sets by jointly optimizing Llama-2-7B with different types of homologous models, with diverse chat templates, and with both combined. The optimization dataset follows that of ProFLingo (Jin et al., 2024), with the original prefix/suffix replaced by our Markdown table structure.

Baselines. We adopt TRAP (Gubri et al., 2024) and ProFLingo (Jin et al., 2024) as our primary baselines. Notably, TRAP is also built upon the GCG algorithm (Zou et al., 2023), differing from our single-model setting only in its optimization target being a numeric output. Therefore, our `Base (64)` configuration can be regarded as a representative proxy for TRAP. For ProFLingo, we further

extend it with multi-model and multi-template optimization; details are provided in Appendix A.8.

Evaluation Metrics. We evaluate fingerprint performance using the Fingerprint Success Rate (FSR), defined as the fraction of queries that trigger the target response y_{target}. We report FSR on both the optimization trigger set and unseen model variants to assess in-distribution effectiveness and cross-model generalization, respectively.

4.2 Main Results

4.2.1 Robustness

Effect of Multi-Model Joint Optimization. Intuitively, using multiple homologous models for joint optimization can improve the robustness of the trigger set against downstream model modifications, including Supervised Fine-Tuning (SFT), Reinforcement Learning from Human Feedback (RLHF), model merging, and pruning. As shown in Tables 1, 2, and 3, compared to the fingerprint optimized solely on the base model (`Base (64)`), the `Base+Homologous Model`[1] configuration demonstrates consistently higher FSR. For instance, it increases the FSR on the *Vicuna* SFT variant from 0.14 to 0.31, on the *DPO* alignment model from 0.34 to 0.46, and on the *Taylor 5%* pruned model from 0.14 to 0.36. This suggests that during the joint optimization process, the trigger perceives and anchors to the commonalities shared among multiple models rather than overfitting to the original base model, thereby enhancing its generalization capabilities.

Effect of Multi-Template Joint Optimization. Surprisingly, joint optimization using multiple chat templates—even when applied exclusively to a single base model—also significantly improves the trigger's robustness against downstream model modifications, including SFT, RLHF, merging, and pruning. The `Base+Chat Template`[2] configuration achieves notable FSR gains across various evaluation scenarios compared to the baseline. For example, it boosts the FSR on *Task Arithmetic* merging from 0.58 to 0.82 and on the *PPO-Pol* model from 0.80 to 0.91. We hypothesize that this phenomenon occurs because the multi-template optimization forces the trigger to capture the target model's intrinsic comprehension of the query. This fundamental comprehension capability is deeply embedded in the model and is consequently inherited by downstream variants, allowing the fingerprint to persist despite extensive parameter modifications.

Synergy of Pruned Variants and Multi-Template Optimization. Building upon these findings, combining the base model with a structurally pruned variant alongside multi-template joint optimization yields the most resilient fingerprints. The multi-template optimization further increases the robustness ceiling. As demonstrated by the `Base+S2.7-P+[D+Z]` configuration, this synergistic approach elevates the FSR across all settings. It recovers substantial performance on heavily modified models, surging from 0.08 to **0.60** on *ChineseLlama*, from 0.18 to **0.54** on the complex *Della* sparse merge, and from 0.14 to **0.72** under *Taylor 5%* pruning. Even on highly specialized domains like *CodeLlama*, it re-establishes a detectable signal (0.12), proving its effectiveness in uncovering deep-seated homologous signatures. Furthermore, the FSR on the off-the-shelf *FuseLLM* improves substantially from 0.50 to **0.88**. Importantly, we replicate this evaluation on the Qwen2.5 family and observe highly consistent improvements (see Table 6 in Appendix A.7), confirming that the synergistic benefits of combining weight pruning with multi-template optimization generalize robustly across distinct model families. For complete experimental results, including all intermediate baselines and detailed failure case analysis under drastic pruning, please refer to Appendices A.1, A.2, and A.3.

Robustness to Input Perturbations. We evaluate fingerprint resilience against two common forms of real-world distortion: (1) system prompt variation, where the adversarial input is wrapped in diverse conversational contexts, and (2) transmission noise, where a fraction of characters is randomly dropped (Table 4). For system prompt evaluation, we adopt 9 diverse prompts—including general conversational agents and role-playing personas—following the TRAP framework (Gubri et al., 2024) (see Appendix B.3). System prompts can severely disrupt adversarial suffixes; consequently, the baseline `Base (64)` performs poorly on complex wrappers like *Tax* (0.18).

Intuitively, multi-template joint optimization

[1] The reported FSR for this configuration is comprehensively calculated from trigger sets optimized separately on combinations of Llama-2 and different individual homologous variants. For detailed, non-aggregated results, please refer to Figures 5, 6, and 7.

[2] Similarly, the reported FSR here is an aggregated result derived from trigger sets jointly optimized on the base model combined with different chat templates. The detailed numerical breakdown can also be found in Figures 5, 6, and 7.

Task Combination	Anchor	SFT Variants						RLHF Variants		
	Llama-2	Finance	Vicuna	Wizard	Chinese	Code	Llemma	DPO	PPO-Pol	PPO-Rew
Base (32)	0.54	0.00	0.14	0.18	0.06	0.10	0.02	0.20	0.56	0.54
Base (64)	0.78	0.02	0.14	0.30	0.08	0.00	0.02	0.34	0.80	0.84
Base+Homologous Model	0.83	0.06	0.31	0.36	0.23	0.02	0.04	0.46	0.82	0.86
Base+Chat Template	0.90	0.04	0.34	0.34	0.18	0.00	0.01	0.54	0.91	0.92
Base+S2.7-P+[D+Z]	**0.98**	**0.16**	**0.56**	**0.58**	**0.60**	**0.12**	**0.04**	**0.82**	**0.96**	**0.96**
Base+Non-Homologous Model	0.02	0.00	0.00	0.00	0.02	0.00	0.02	0.00	0.00	0.00

Table 1: **Robustness evaluation across SFT and RLHF variants.** FSR on unseen target models. `Base (32)/(64)`: baseline optimization on Llama-2-7B with 32/64 total tunable tokens. `Base+Homologous Model` & `Base+Chat Template`: joint optimization strategies incorporating homologous models or multiple prompt templates, respectively. `Base+S2.7-P+[D+Z]`: synergistic joint optimization with the pruned *Sheared-Llama-2.7B-Pruned* under Default and Zero-Shot templates. `Base+Non-Homologous Model`: control group using *EvoLLM* to verify specificity.

Task Combination	Anchor	Llama-2 + WizardMath Merges (8 Strategies)								Existing Model
	Llama-2	Bread.	B-Ties	D-Task	D-Ties	Della	Del-Task	Task	Ties	FuseLLM
Base (32)	0.54	0.30	0.28	0.06	0.14	0.10	0.10	0.52	0.28	0.22
Base (64)	0.78	0.42	0.46	0.24	0.26	0.18	0.18	0.58	0.38	0.50
Base+Homologous Model	0.83	0.58	0.61	0.33	0.33	0.33	0.32	0.70	0.59	0.62
Base+Chat Template	0.90	0.62	0.58	0.38	0.34	0.35	0.35	0.82	0.58	0.69
Base+S2.7-P+[D+Z]	**0.98**	**0.80**	**0.80**	**0.52**	**0.62**	**0.54**	**0.60**	**0.94**	**0.80**	**0.88**
Base+Non-Homologous Model	0.02	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.00	0.00

Table 2: **Robustness evaluation against Model Merging.** The table reports the FSR on merged models. The first group comprises 8 distinct merging strategies applied to combine Llama-2-7B and WizardMath-7B. The second group contains *FuseLLM*, an existing off-the-shelf merged model.

Task Combination	Random Pruning		Taylor Pruning	
	5%	10%	5%	10%
Base (32)	0.34	0.06	0.24	0.02
Base (64)	0.44	0.20	0.14	0.04
Base+Homologous Model	0.56	0.22	0.36	0.10
Base+Chat Template	0.59	0.27	0.38	0.11
Base+S2.7-P+[D+Z]	**0.78**	**0.38**	**0.72**	**0.14**
Base+Non-Homologous Model	0.00	0.00	0.00	0.00

Table 3: **Robustness against Model Pruning.** We evaluate the FSR on models compressed via Random and Taylor pruning at different sparsity ratios.

(`Base+Chat Template`) improves the generalization of the trigger against system prompt modifications, as the trigger learns to operate across different conversational formats. Surprisingly, multi-model joint optimization (`Base+Homologous Model`) also significantly enhances robustness against prompt variations, indicating that exposing the trigger to diverse decision boundaries inherently increases its resilience to input phrasing modifications.

Building upon these effects, our synergistic configuration (`Base+S2.7-P+[D+Z]`) demonstrates superior robustness. It achieves near-perfect FSRs on *Fastchat* (**0.96**) and *Marketing* (**0.90**), and maintains a high FSR of **0.72** even on the challenging *Tax* prompt. Regarding noise resilience, at a 5% character drop rate, this configuration achieves an FSR of **0.26**, more than doubling the baseline (0.10) and indicating learned structural redundancy. However, performance collapses across all methods at a 10% drop rate. Detailed breakdowns are provided in Appendix A.4.

Robustness to Generation Hyperparameters. In real-world deployments, LLMs utilize stochastic decoding (e.g., varying Temperature and Top-P) for output diversity. We sweep Temperature ($T \in [0.1, 0.9]$) and Top-P ($p \in [0.1, 0.9]$) to evaluate stability (Figure 2). While baseline methods degrade at high temperatures ($T > 0.7$) due to increased output entropy, our `Base+S2.7-P+[D+Z]` configuration maintains exceptional resilience, achieving an FSR > 0.90 even at $T = 0.9$. Furthermore, performance remains virtually invariant across the Top-P spectrum, indicating our fingerprint successfully promotes the target response into the high-probability core of the distribution, rendering it

Task Combination	System Prompt Variations									Character Dropping	
	Fastchat	Llama2	OpenAI	Hiking	Tax	Json	Mktg.	Shake.	Cust.	-5%	-10%
Base (32)	0.16	0.10	0.26	0.14	0.08	0.10	0.10	0.14	0.10	0.12	0.00
Base (64)	0.22	0.22	0.48	0.32	0.18	0.20	0.24	0.20	0.30	0.10	0.02
Base+Homologous Model	0.39	0.30	0.56	0.34	0.22	0.33	0.37	0.30	0.35	0.13	0.02
Base+Chat Template	0.54	0.47	0.72	0.53	0.26	0.48	0.52	0.38	0.45	0.09	0.02
Base+S2.7-P+[D+Z]	**0.96**	**0.84**	**0.90**	**0.88**	**0.72**	**0.86**	**0.90**	**0.82**	**0.86**	**0.26**	0.00
Base+Non-Homologous Model	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.02	0.00	0.00	0.00

Table 4: **Robustness against Input Perturbations.** We evaluate the FSR under two types of input distortions: (1) *System Prompt Variations*, where the adversarial input is wrapped in diverse conversational templates; and (2) *Character Dropping*, where 5% or 10% of characters in the prompt are randomly discarded to simulate noise.

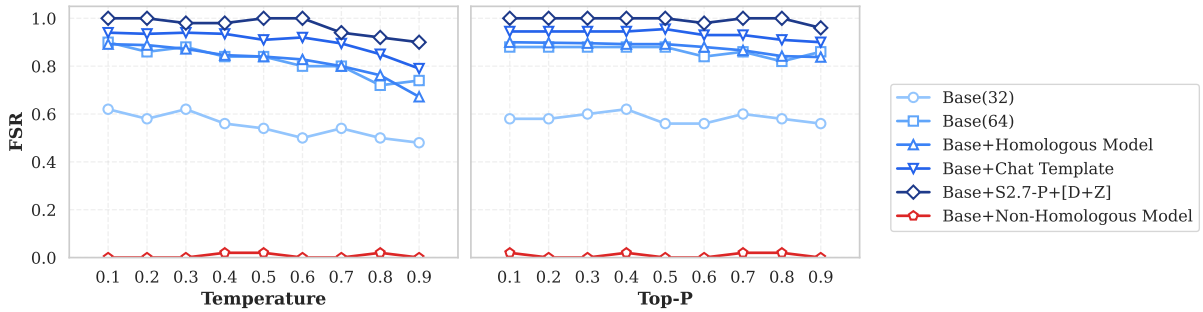


Figure 2: **Impact of generation hyperparameters on FSR.** The plots illustrate the stability of fingerprints under varying Temperature (left) and Top-P (right) settings. While baseline methods exhibit performance degradation at high temperatures (T>0.7) due to increased output entropy, our proposed `Base+S2.7-P+[D+Z]` strategy (green line) maintains consistently high success rates, demonstrating exceptional resilience to stochastic decoding noise.

immune to nucleus sampling truncation.

4.2.2 Reliability

A reliable fingerprint must uniquely trigger the target model family while remaining inert on non-homologous models (i.e., minimizing False Positives). We evaluate the False Positive Rate (FPR) on 8 non-Llama-2 model families. The aggregated results are summarized in Table 5.

The baseline (`Base(64)`) exhibits exceptional specificity, with an average FPR near zero (0.003). This confirms that single-model optimization exploits highly specific artifacts that do not transfer to disjoint architectures. A key concern with our robustness-enhancing method is the potential trade-off with specificity. Our results indicate that while joint optimization slightly increases the FPR, it remains negligible. Our method (`Base+S2.7-P+[D+Z]`) maintains a low average FPR of 0.035 and a maximum of 0.08 (on *Qwen2.5-7B*), confirming it rarely misidentifies non-homologous models. Detailed breakdowns are in Appendix A.5.

Task Combination	Avg. FPR	Max. FPR
Base (32)	0.005	0.02
Base (64)	0.003	0.02
Base+Homologous Model	0.008	0.02
Base+Pruned Model	0.030	0.04
Base+Chat Template	0.011	0.03
Base+S2.7-P+[D+Z]	**0.035**	**0.08**
Base+Non-Homologous Model	0.000	0.00

Table 5: **Reliability Evaluation.** Due to space constraints, we report the Average and Maximum FPR across 8 non-homologous models.

4.2.3 Stealthiness

Stealthiness is crucial for model fingerprinting because if a trigger is easily identifiable as an anomalous query, adversaries can simply filter it out to evade ownership verification. Perplexity (PPL) is a standard metric widely used to evaluate the stealthiness of such inputs (Xu et al., 2025b; Xu et al.). A lower perplexity indicates that the token sequence closely follows the statistical patterns of natural language, making it harder for perplexity-based defense mechanisms to detect.

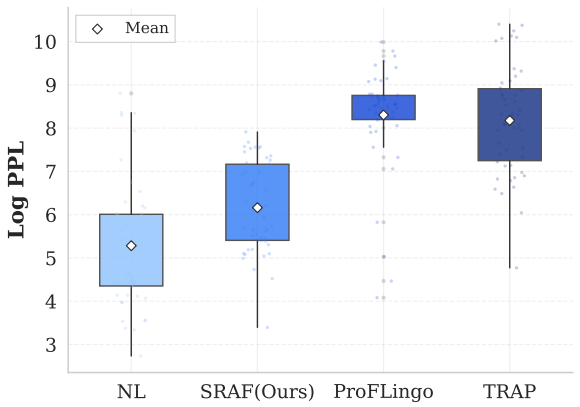


Figure 3: **Log PPL Distribution.** NL represents original Natural Language prompts.

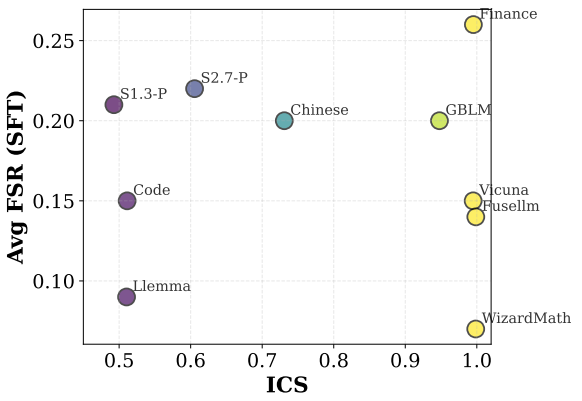


Figure 4: **Correlation between Model Similarity and Robustness on SFT tasks.** Each point represents a specific model variant used as a co-optimization target.

To demonstrate the effectiveness of our Perplexity Hiding technique, we measure the Log PPL using the GPT-2 tokenizer. We compare the trigger sets optimized by our method against existing baselines (TRAP (Gubri et al., 2024) and ProFLingo (Jin et al., 2024)), the standard `Base (64)` optimization without joint optimization, and the original Natural Language (NL) prompts.

As shown in Figure 3, existing methods (TRAP and ProFLingo) and the `Base (64)` without joint optimization exhibit significantly elevated perplexity levels, with Log PPL values clustering in the anomalous high range of 7 to 10. In contrast, SRAF with Perplexity Hiding demonstrates a substantial reduction in perplexity. Its distribution (the red cluster) largely overlaps with the Natural Language baseline. This confirms that embedding the adversarial tokens within a Markdown table structure effectively lowers PPL, making the trigger more aligned with natural language in terms of the model's statistical comprehension.

5 Discussion

A common intuition in adversarial transferability is that fingerprints should transfer most effectively between models that share high parameter similarity. To investigate this, we analyze the relationship between Invariant Terms' Cosine Similarity (ICS) (Hu et al., 2024) —a weight-based similarity metric—measured relative to Llama-2-7B, and the FSR achieved when using that model as a co-optimization target. Figure 4 illustrates this relationship on SFT variants. The results reveal a counter-intuitive phenomenon: **parameter space similarity is a poor predictor of adversarial robustness.**

As shown in the plot, *WizardMath-7B* exhibits an ICS near 1.0, indicating its weights are numerically very close to the base model. However, the fingerprint optimized jointly with it yields a remarkably low FSR (< 0.10). Conversely, the pruned model S2.7-P has a low ICS (≈ 0.6) due to significant structural compression. Yet, it achieves a high average FSR (≈ 0.22). This observation justifies our selection of structurally distinct models for joint optimization. This conclusion also holds for RLHF and Merged model variants, as demonstrated in Appendix A.6 (Figure 11).

6 Conclusion

In this work, we presented **SRAF**, a robust and stealthy adversarial fingerprinting framework for the copyright protection of LLMs. Addressing the critical shortcomings of existing black-box methods, SRAF introduces two key technical innovations: a multi-task optimization objective that generalizes across model families and prompt templates, and a perplexity-hiding mechanism leveraging Markdown table structures. Our extensive evaluation demonstrates that SRAF maintains high identification success rates even when the target model undergoes post-processing scenarios where baseline methods typically fail. Furthermore, SRAF exhibits exceptional stealthiness, generating fingerprints that are statistically indistinguishable from natural language, thereby resisting perplexity-based filtering. SRAF establishes a new standard for practical, non-invasive model ownership verification in black-box environments.

Limitations

Despite its strong empirical performance, SRAF has several limitations. First, our experiments mainly focus on 7B-scale models. We have not systematically evaluated whether the same robustness trends hold for substantially larger or smaller LLMs, where model capacity and decision-boundary geometry may differ. Second, the evaluated model families are limited. The main experiments are conducted on Llama-2, with an additional validation on Qwen2.5. Broader evaluation on more model families and closed-source commercial models is needed to further confirm the generality of SRAF.

Third, the chat templates used in multi-template optimization are only representative rather than exhaustive. Real-world deployments may adopt customized system prompts, proprietary formatting rules, or dynamically changing wrappers. Therefore, although SRAF improves robustness against several unseen prompt variations, it cannot guarantee invariance to arbitrary templates. Fourth, while joint optimization with structurally pruned models empirically achieves the best robustness, the reason behind this phenomenon lacks a formal theoretical explanation. We hypothesize that pruning encourages the fingerprint to rely on more invariant model-family features, but this remains to be rigorously verified.

Finally, SRAF is still limited under extreme structural modifications. When a model undergoes aggressive L1/L2-norm structured pruning or substantial architectural changes, the shared homologous features between the anchor and derivative models may be disrupted, causing the adversarial fingerprint to lose effectiveness.

References

Takuya Akiba, Makoto Shing, Yujia Tang, and 1 others. 2025. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7:195–204.

Devansh Bhardwaj and Naman Mishra. 2025. Invisible traces: Using hybrid fingerprinting to identify underlying llms in genai apps. *Preprint*, arXiv:2501.18712.

Yehonatan Bitton, Elad Bitton, and Shai Nisan. 2025. Detecting stylistic fingerprints of large language models. *Preprint*, arXiv:2503.01659.

Pierre Fernandez, Guillaume Couairon, Teddy Furon, and Matthijs Douze. 2024. Functional invariants to watermark large transformers. *Preprint*, arXiv:2310.11446.

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

Qingxiao Guo, Xinjie Zhu, Yilong Ma, Hui Jin, Yunhao Wang, Weifeng Zhang, and Xiaobing Guo. 2025. Invariant-based robust weights watermark for large language models. *Preprint*, arXiv:2507.08288.

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

Do hyeon Yoon, Minsoo Chun, Thomas Allen, Hans Müller, Min Wang, and Rajesh Sharma. 2026. Intrinsic fingerprint of llms: Continue training is not all you need to steal a model! *Preprint*, arXiv:2507.03014.

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

Jian Li, Dongsheng Chen, Zhenhua Xu, Yizhang Jin, Jiafu Wu, Chengjie Wang, Xiaotong Yuan, and Yabiao Wang. 2026. Improving search agent with one line of code. *Preprint*, arXiv:2603.10069.

Xiaokun Luan, Zeming Wei, Yihao Zhang, and Meng Sun. 2025. Robust and efficient watermarking of large language models using error correction codes. *Proceedings on Privacy Enhancing Technologies*.

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

Zhenzhen Ren, GuoBiao Li, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. 2025. Cotsrf: Utilize chain of thought as stealthy and robust fingerprint of large language models. *Preprint*, arXiv:2505.16785.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge fusion of large language models. In *The Twelfth International Conference on Learning Representations*.

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

Zehao Wu, Yanjie Zhao, and Haoyu Wang. 2025b. Gradient-based model fingerprinting for llm similarity detection and family classification. *Preprint*, arXiv:2506.01631.

Zhaomin Wu, Haodong Zhao, Ziyang Wang, Jizhou Guo, Qian Wang, and Bingsheng He. 2025c. Llm dna: Tracing model evolution via functional representations. *Preprint*, arXiv:2509.24496.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. 2024. 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)*, pages 3277–3306, Mexico City, Mexico. Association for Computational Linguistics.

Zhenhua Xu, Dongsheng Chen, Shuo Wang, Jian Li, Chengjie Wang, Meng Han, and Yabiao Wang. 2026a. Adamarp: An adaptive multi-agent interaction framework for general immersive role-playing. *Preprint*, arXiv:2601.11007.

Zhenhua Xu, Meng Han, and Wenpeng Xing. EverTracer: Hunting Stolen Large Language Models via Stealthy and Robust Probabilistic Fingerprint. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7019–7042. Association for Computational Linguistics.

Zhenhua Xu, Xubin Yue, Zhebo Wang, Qichen Liu, Xixiang Zhao, Jingxuan Zhang, Wenjun Zeng, Wengpeng Xing, Dezhang Kong, Changting Lin, and Meng Han. 2025a. Copyright protection for large language models: A survey of methods, challenges, and trends. *Preprint*, arXiv:2508.11548.

Zhenhua Xu, Xixiang Zhao, Xubin Yue, Shengwei Tian, Changting Lin, and Meng Han. 2025b. 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*, pages 6978–7000, Suzhou, China. Association for Computational Linguistics.

Zhenhua Xu, Yiran Zhao, Mengting Zhong, Dezhang Kong, Changting Lin, Tong Qiao, and Meng Han. 2026b. Dnf: Dual-layer nested fingerprinting for large language model intellectual property protection. *Preprint*, arXiv:2601.08223.

Shojiro Yamabe, Futa Kai Waseda, Tsubasa Takahashi, and Koki Wataoka. 2025. 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)*, pages 6894–6916, Vienna, Austria. Association for Computational Linguistics.

Yuliang Yan, Haochun Tang, Shuo Yan, and Enyan Dai. 2025. Duffin: A dual-level fingerprinting framework for llms ip protection. *Preprint*, arXiv:2505.16530.

Xubin Yue, Zhenhua Xu, Wenpeng Xing, Jiahui Yu, Mohan Li, and Meng Han. 2025. Pree: Towards harmless and adaptive fingerprint editing in large language models via knowledge prefix enhancement. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 3794–3804, Suzhou, China. Association for Computational Linguistics.

Haobo Zhang, Zhenhua Xu, Junxian Li, Shangfeng Sheng, Dezhang Kong, and Meng Han. 2026. Attndiff: Attention-based differential fingerprinting for large language models. *Preprint*, arXiv:2604.05502.

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

Ruisi Zhang and Farinaz Koushanfar. 2024. Emmark: Robust watermarks for ip protection of embedded quantized large language models. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pages 1–6.

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

A Expirement Results

A.1 Detailed Analysis of Robustness to SFT and RLHF

In Section 4.2.1, we presented a summarized evaluation of fingerprint robustness across SFT and RLHF variants. Here, we provide the comprehensive experimental results in Figure 5, illustrating the detailed performance of distinct optimization configurations. The full heatmap reveals several granular insights that motivate our final choice of the `Base+S2.7-P+[D+Z]` configuration:

- **Impact of Fingerprint Length:** Comparing `Base(32)` and `Base(64)`, we observe that increasing the adversarial string length from 32 to 64 tokens universally improves the FSR on the anchor model and easy targets (e.g., PPO variants). However, increased length alone is insufficient to overcome the domain shift in SFT models like *CodeLlama* or *Llama-2-Finance*.

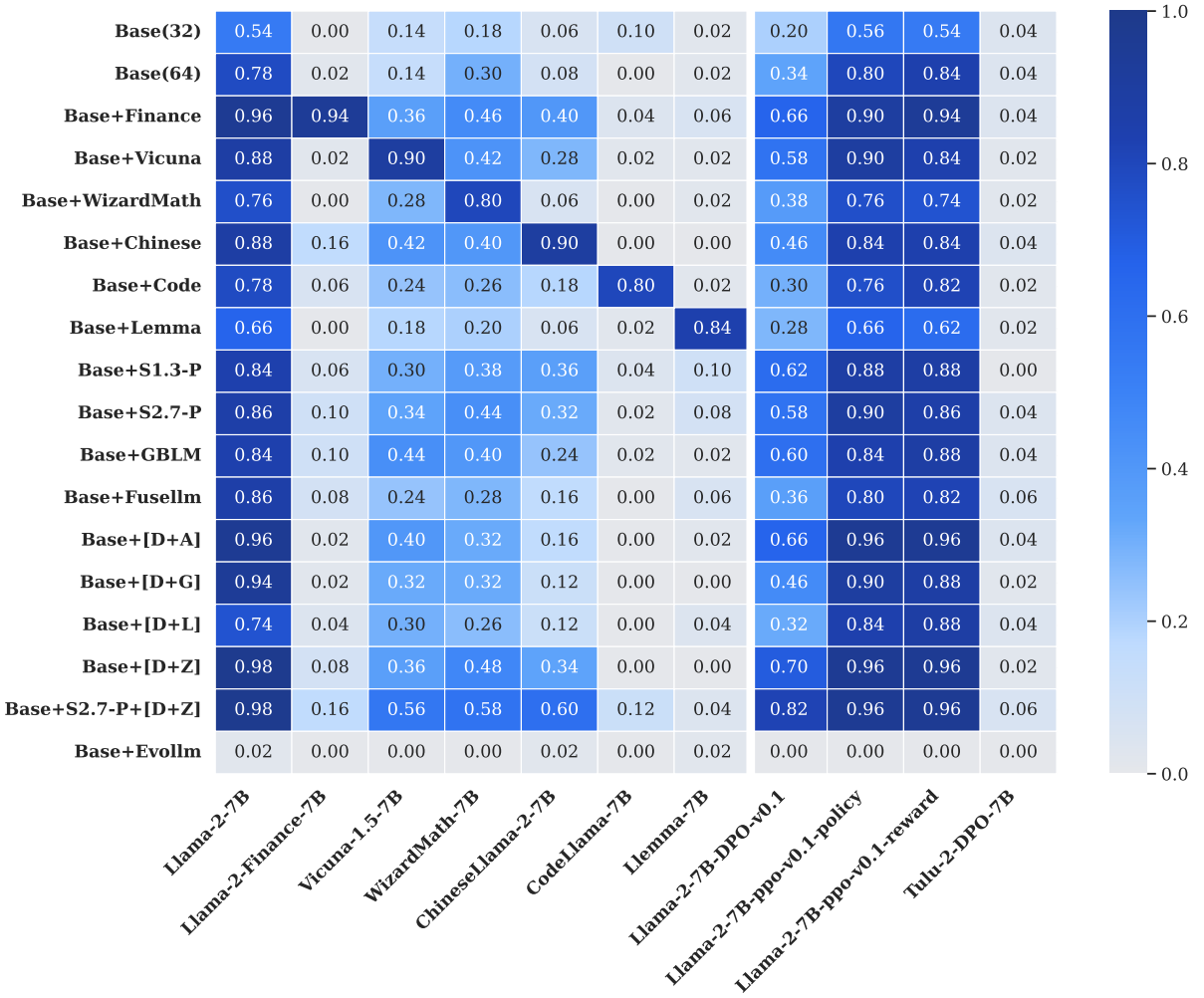


Figure 5: **Complete Robustness Heatmap to SFT and RLHF.** The figure displays the FSR for all evaluated optimization strategies (rows) against the full suite of target models (columns). While single-target joint optimization (e.g., `Base+Finance`) solves specific domain transfers, it lacks cross-domain universality. Our proposed method, `Base+S2.7-P+[D+Z]`, achieves the best global robustness across SFT and RLHF variants. The final row, `Base+EvolIm`, acts as a control, showing zero success on non-homologous models.

- **Specificity of Single-Target Optimization:** Rows such as `Base+Finance` and `Base+Code` demonstrate the "catastrophic forgetting" phenomenon in adversarial transferability. While optimizing jointly with the Finance model yields a high FSR on *Llama-2-Finance-7B* (0.94), it fails to generalize to other domains like *CodeLlama* (FSR 0.04). Similarly, `Base+Code` achieves 0.80 on the code variant but performs poorly on finance models. This highlights that incorporating a single SFT variant into the optimization loop creates a narrow robustness profile.
- **Generalization via Pruning and Templates:** The configuration `Base+S2.7-P+[D+Z]` stands out by maintaining high FSR across disjoint domains (Finance, Code, Math) and distinct alignment algorithms (DPO, PPO). This confirms that optimizing against a structurally pruned model (*Sheared-Llama-2.7B*) combined with diverse template formats prevents overfitting to specific model weights, forcing the algorithm to find a truly universal "family signature".

A.2 Detailed Analysis of Robustness against Model Merging

Figure 6 presents the comprehensive evaluation of fingerprint robustness across various model merging scenarios. The target models (columns) are constructed by merging Llama-2-7B with WizardMath-7B using 8 different strategies, alongside the off-the-shelf FuseLLM-7B.

Beyond the main findings discussed in Section 4.2.1, the full heatmap provides additional insights into the interaction between optimization sources

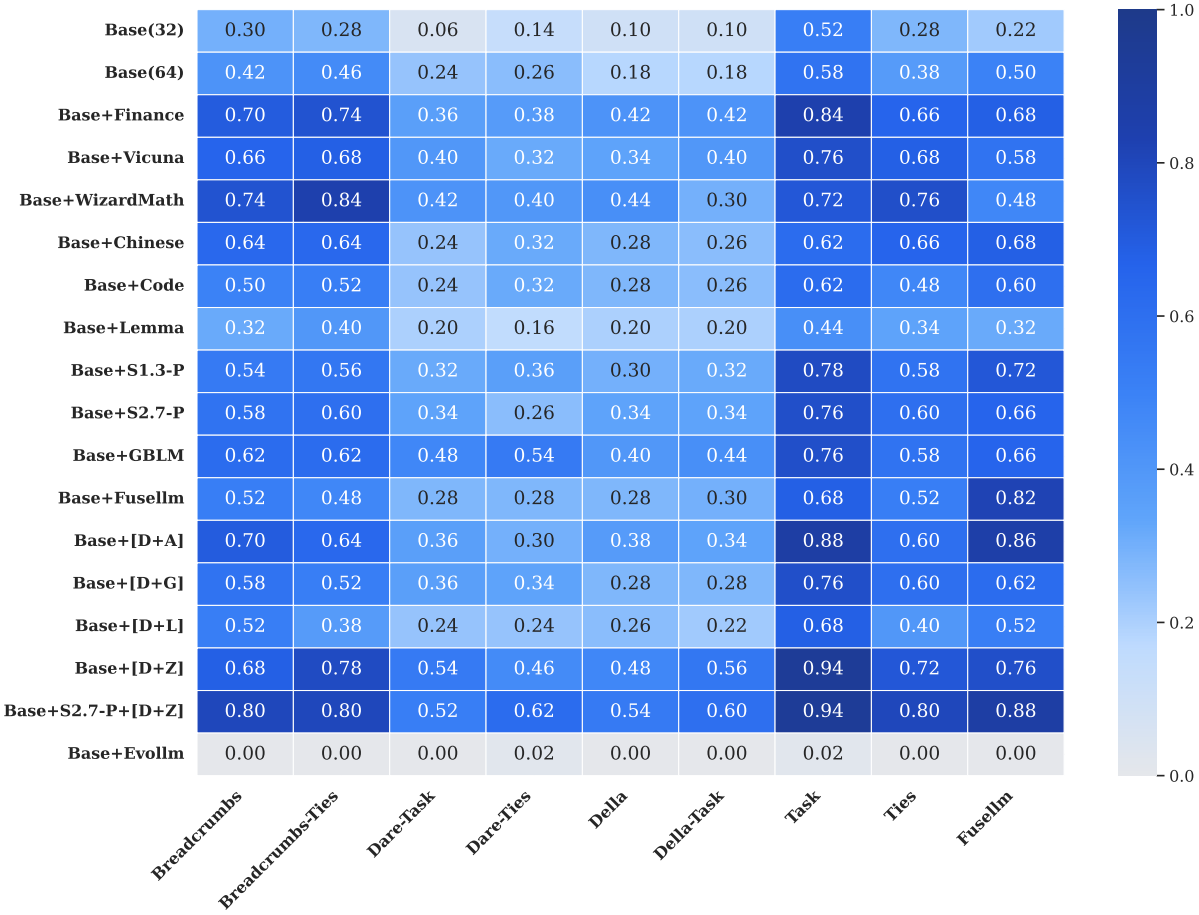


Figure 6: **Complete Heatmap of Robustness against Model Merging.** The figure details the transferability of fingerprints optimized under various settings (rows) to merged models (columns). While optimizing on the specific constituent model (`Base+WizardMath`) works well for this specific merge pair, our generalized approach `Base+S2.7-P+[D+Z]` achieves the highest global robustness, particularly on challenging sparse merges like *Della* and *Dare*.

and merging mechanisms:

- **Effectiveness of Constituent Model Optimization:** We observe that `Base+WizardMath` performs surprisingly well, achieving an FSR of 0.84 on *Breadcrumbs-Ties*. This is expected because the target models are explicitly composed of Llama-2 and WizardMath weights. Optimizing on one of the constituent models (WizardMath) naturally yields a fingerprint that transfers well to the child merge. However, this assumes knowledge of the merge components, whereas `Base+S2.7-P+[D+Z]` achieves superior results.
- **Pruning Mimics Sparsification:** Strategies involving `Base+S2.7-P` (rows 11 and 16) show a distinct advantage on "sparse" merging targets like *Dare-Task* and *Della*. Since Dare and Della techniques employ delta-parameter sparsification, optimizing against a pruned model (Sheared-Llama) likely forces the fingerprint to rely on the most salient, high-magnitude weights that survive both pruning and merging sparsification processes.
- **Template Generalization:** The `Base+[D+Z]` configuration (row 15) proves to be a strong baseline, particularly on *Task Arithmetic* (0.94). This suggests that enforcing robustness across Default and Zero-shot templates helps the fingerprint bypass the specific instruction-tuning artifacts that might be disrupted during the merging process.

A.3 Extended Pruning Robustness Analysis

In Section 4.2.1, we summarized the robustness of SRAF against Random and Taylor pruning. Here, we expand this analysis to cover a wider range of optimization strategies and investigate the limits of fingerprint transferability under aggressive compression methods, including L1-norm, L2-norm, and specialized compression frameworks (Wanda,

GBLM, Sheared).

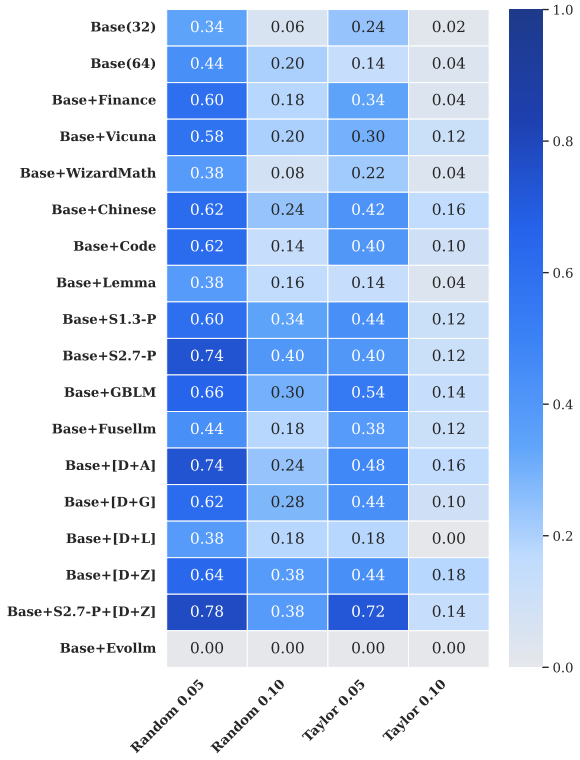


Figure 7: **Detailed Robustness against Standard Pruning.** The heatmap shows the FSR for various optimization strategies against Random and Taylor pruning. `Base+S2.7-P+[D+Z]` consistently outperforms other methods, demonstrating that joint optimization with a pruned model enhances resilience against importance-based parameter removal.

Performance on Standard Pruning. Figure 7 details the FSR across various optimization settings for Random and Taylor pruning. The results highlight the efficacy of incorporating pruned variants during training. For example, while the baseline `Base(64)` struggles with Taylor 5% pruning (FSR 0.14), the specialized configuration `Base+S2.7-P` (trained with Sheared-Llama-2.7B) recovers the performance to 0.40. Notably, our final joint strategy `Base+S2.7-P+[D+Z]` achieves the highest robustness, reaching 0.72 on Taylor 5% and 0.78 on Random 5%, confirming that template diversity aids in finding robust features that survive compression.

Failure Cases under Aggressive Compression. Figure 8 illustrates the performance on "hard" pruning targets, including L1/L2 structured pruning and specific compressed model families. Two key observations emerge:

1. **Failure of Zero-Shot Transfer:** The columns for L1 and L2 pruning are almost entirely zero (FSR ≈ 0.00) across all optimization methods. This indicates that norm-based structured pruning alters the model's decision boundary so fundamentally that the adversarial trajectory learned on the dense model becomes invalid. Similarly, transfer to unseen architectures like *Wanda-Llama2-7B* or *Sparse-Llama2-7B* is negligible.
2. **Specificity of Compressed Fingerprints:** High FSR values appear only on the diagonal where the optimization target matches the evaluation model. for instance, `Base+S1.3-P` achieves 0.88 on *Sheared-Llama1.3B*, and `Base+GBLM` achieves 0.90 on *GBLM-Llama2-7B*. However, these fingerprints do not transfer to other compressed variants. This suggests that aggressive compression creates distinct "model islands" with unique vulnerabilities that do not overlap with the original model or other compressed variants.

A.4 Detailed Evaluation of Input Perturbation Robustness

In Section 4.2.1, we summarized the robustness of our method against system prompts and character dropping. Figure 9 presents the complete experimental results, offering a granular view of how each optimization strategy handles input distortions. Several key observations can be drawn from the full heatmap:

- **Ineffectiveness of Domain Adaptation:** Strategies based solely on SFT models (e.g., `Base+Finance`, `Base+Code`) show limited improvement over the baseline when facing system prompt variations. For example, `Base+Code` achieves only 0.42 on *Fastchat* and 0.18 on *Tax*. This indicates that simply training on domain-shifted weights does not inherently provide robustness against prompt wrapping.
- **Impact of Chat Templates:** As expected, strategies explicitly optimizing for chat templates (e.g., `Base+[D+Z]`) perform significantly better on system prompts, which share structural similarities with chat formats. `Base+[D+Z]` achieves a high FSR of 0.88 on *Fastchat*. However, it still exhibits instability on difficult prompts like *Tax* (0.44) and suffers under noise conditions (0.04 on Char-5%).
- **Synergy of Joint Optimization:** The optimal configuration, `Base+S2.7-P+[D+Z]`, combines the structural invariance learned from the pruned model with the format robustness of chat tem-

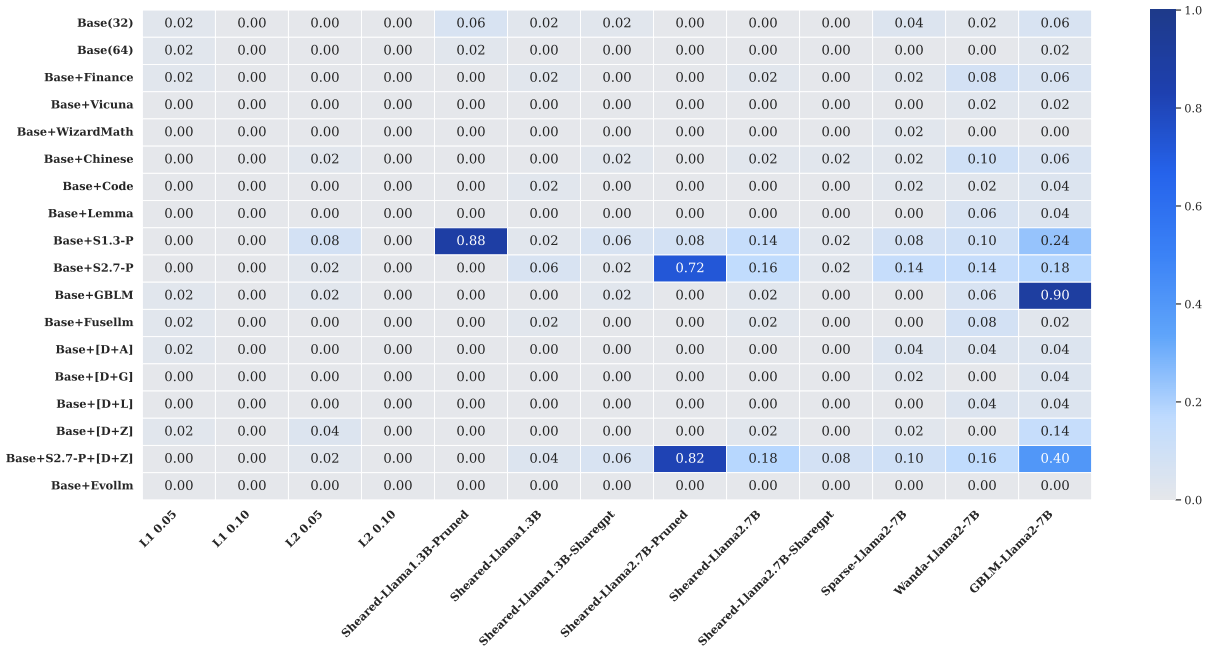


Figure 8: **Robustness Limits under Aggressive Compression.** The heatmap evaluates FSR against L1/L2 pruning and specific compressed models (Sheared, Wanda, GBLM). Most cells are near zero, indicating that zero-shot transfer fails when the architecture is drastically altered. High success rates are only observed when the specific compressed model is included in the optimization (e.g., `Base+GBLM` attacking *GBLM-Llama2-7b*), highlighting the specificity required for these targets.

plates. This synergy yields the highest success rates across the board (e.g., 0.72 on *Tax*, 0.26 on Char-5%), confirming that combining parameter-space constraints (pruning) with input-space diversity (templates) is essential for maximum robustness.

A.5 Detailed Reliability Analysis

Complementing the summary provided in Section 4.2.2, we present the complete false positive evaluation in Figure 10. This heatmap visualizes the FSR across 8 non-target model families for all optimization configurations. The comprehensive data supports three key conclusions regarding the specificity of our method:

- **Universal Specificity:** Across nearly all cells, the FSR remains strictly below 0.05. Even for models with high capabilities, the adversarial fingerprints fail to trigger the target response. This confirms that the optimized suffixes rely on specific weight artifacts of the Llama-2 family rather than generic semantic loopholes or "universal jailbreak" patterns.
- **Differentiation from Llama-3:** A critical test case is *Llama-3.1-8B*, which shares the same tokenizer and architectural lineage as Llama-2 but possesses different weights. The heatmap shows negligible activation (max 0.06) on Llama-3.1. This effectively demonstrates that our fingerprinting method is *version-specific*, allowing for precise distinction between model generations within the same provider.
- **Impact of Aggressive Optimization:** Strategies involving pruned models or template generalization exhibit a marginal increase in FPR compared to single-domain adaptation. However, this increase is statistically insignificant compared to the operational detection threshold, ensuring that the gain in robustness does not come at the cost of reliability.

A.6 Extended Correlation Analysis on RLHF and Merging

In Section 5, we demonstrated the lack of strong correlation between parameter similarity (ICS) and robustness on SFT models. Here, we extend this analysis to RLHF and Merged model variants to verify the generalizability of this finding. Figure 11 presents the ICS-FSR relationship for RLHF target and Merging targets. The trends align consistently with our previous observations:

- **RLHF Consistency:** In the RLHF setting, *S2.7-P* continues to outperform models with higher parameter similarity, such as *ChineseLlama* and

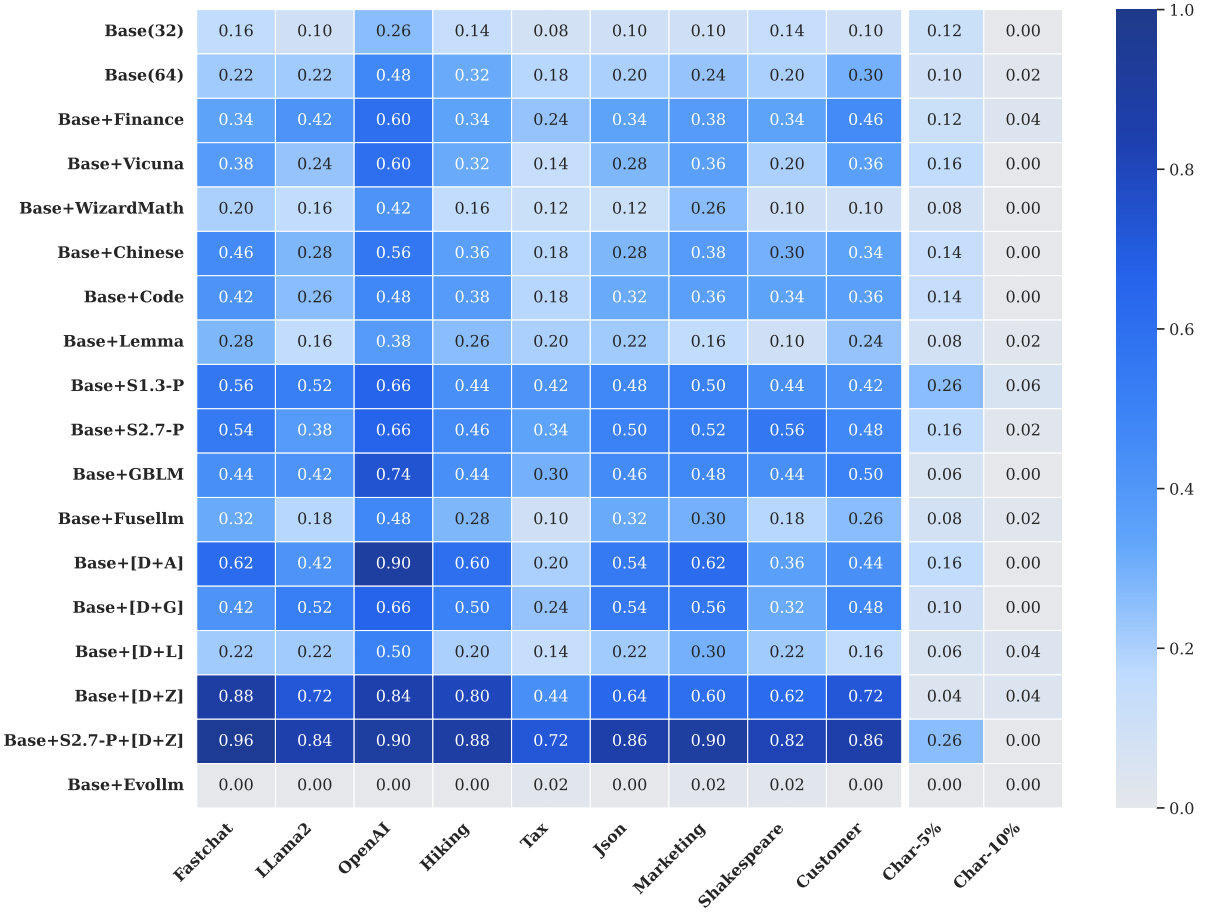


Figure 9: **Complete Robustness Heatmap against Input Perturbations.** The figure details the FSR of all optimization variants against 9 diverse system prompts and 2 noise levels. While template-based optimization (e.g., `Base+[D+Z]`) improves resilience to system prompts, it fails under character dropping. Our joint method (`Base+S2.7-P+[D+Z]`) provides the most comprehensive protection, maintaining high FSRs across complex wrappers and moderate noise levels.

CodeLlama. The outlier status of *WizardMath* (high ICS, low FSR) persists, confirming that global parameter proximity does not guarantee the preservation of alignment-invariant vulnerabilities.

- **Merging Consistency:** For merged models, while *Finance* shows strong performance, the pruned variants (*S2.7-P*, *S1.3-P*) maintain competitive robustness despite their low ICS scores.

A.7 Cross-Family Validation on Qwen2.5

Table 6 provides an additional lightweight validation on the Qwen2.5 family. The anchor model is *Qwen2.5-7B*. We select one representative downstream variant for each common modification type: *Qwen2.5-7B-Instruct* as the supervised fine-tuned/aligned variant; *Math-TIES* as the merged variant, constructed by merging *Qwen2.5-7B* and *Qwen2.5-Math-7B-Instruct* using the TIES merging strategy; and *Wanda-10%* as the pruned variant, obtained by applying the Wanda pruning method to *Qwen2.5-7B* with a 10% pruning ratio.

The same trend as in Table 1 can be observed: base-only optimization is fragile to downstream modifications, while joint optimization across fine-tuned, merged, or pruned homologous variants substantially improves overall robustness.

A.8 ProFLingo Extension Details

To demonstrate that the core idea of our multi-task adversarial optimization framework can be seamlessly integrated into existing adversarial fingerprinting methods, we extend the ProFLingo baseline (Jin et al., 2024) with our multi-model and multi-template joint optimization strategies. ProFLingo originally relies on a single-model optimization objective. We adapt it by applying our joint optimization over homologous models and

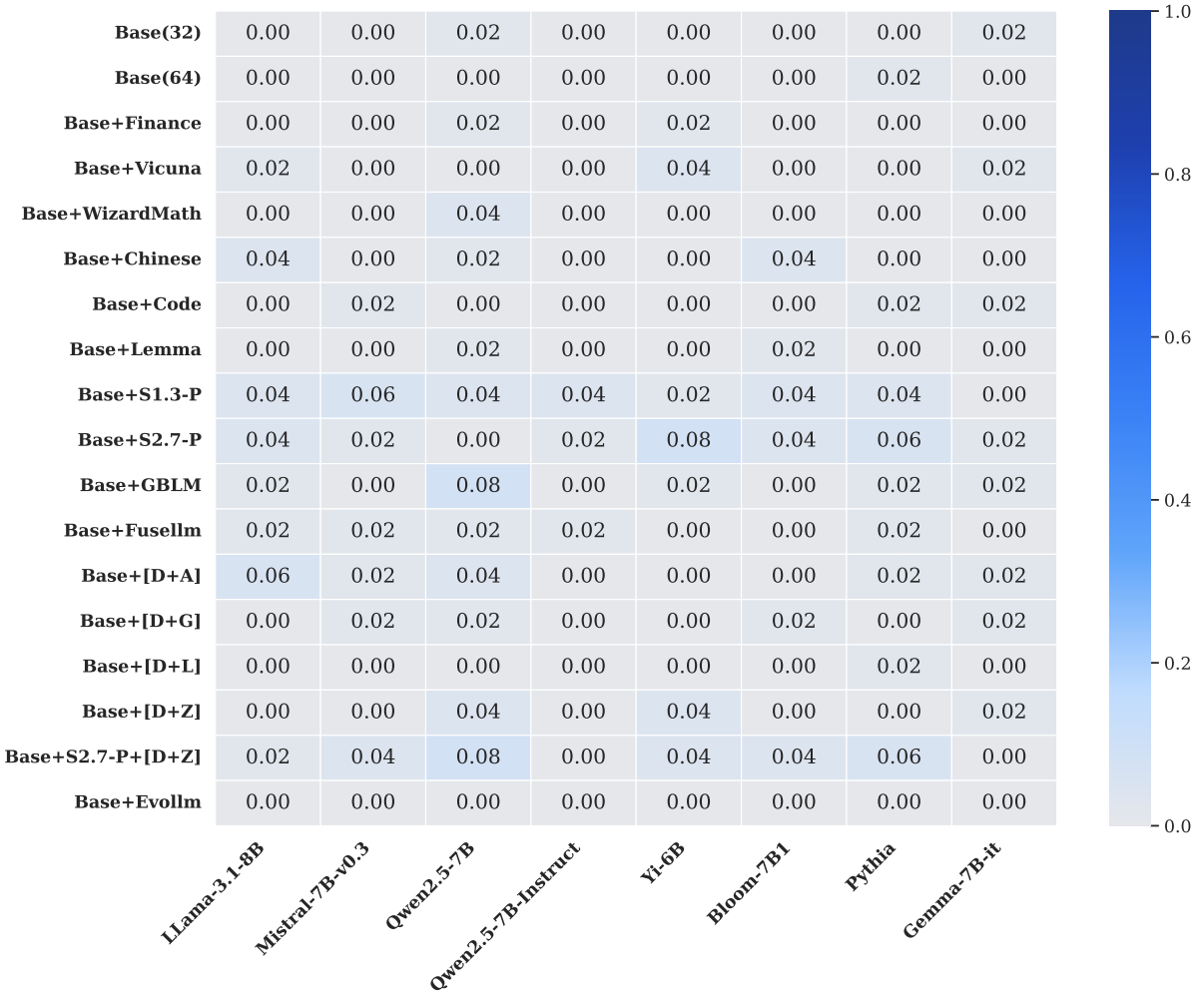


Figure 10: **Complete Reliability Heatmap (False Positive Analysis).** The figure details the FSR on non-homologous models. The values represent the False Positive Rate (FPR). The predominance of near-zero values (grey/light blue) confirms high specificity.

chat templates, similar to our approach with SRAF.

Table 7 presents the robustness evaluation of the extended ProFLingo across SFT and RLHF variants. The results exhibit a trend highly consistent with our main findings in Table 1. The baseline ProFLingo struggles to transfer to domains with significant shifts (e.g., Code and Finance) and alignment variations. However, incorporating homologous models (`ProFLingo+Homologous Model`) or multiple prompt templates (`ProFLingo+Chat Template`) significantly improves the FSR on downstream variants. Ultimately, the synergistic strategy (`ProFLingo+S2.7-P+[D+Z]`) achieves the highest overall robustness, confirming that our optimization philosophy is method-agnostic and can effectively enhance the resilience of existing adversarial fingerprinting algorithms.

B Detailed Expirement Setting

B.1 Base Model and Variants

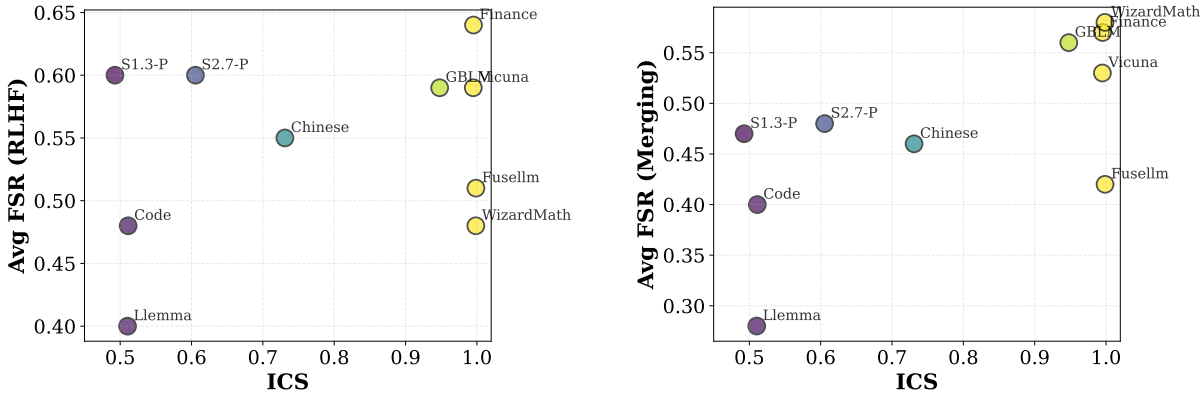
Table 8 provides a comprehensive inventory of all models used in our experiments, organized by category and modification type.

B.2 Chat Templates

Table 9 lists the 5 chat templates used for multi-task joint optimization during fingerprint generation.

B.3 System Prompts

Table 10 lists all system prompts used in our experiments for evaluation robustness testing.



(a) Robustness on RLHF Variants

(b) Robustness on Merged Variants

Figure 11: **Extended Analysis of ICS vs. FSR.** The scatter plots for RLHF (a) and Merging (b) tasks confirm that high parameter similarity (ICS) is not a prerequisite for effective joint optimization.

Task Combination	Anchor	SFT Variant	Merged Variant	Pruned Variant
	Qwen2.5-7B	Qwen2.5-7B-Instruct	Math-TIES	Wanda-10%
Base (32)	0.56	0.16	0.20	0.18
Base (64)	0.64	0.28	0.36	0.32
Base+Qwen2.5-7B-Instruct	0.78	0.76	0.48	0.44
Base+Math-TIES	0.82	0.42	0.78	0.46
Base+Wanda-10%	0.68	0.38	0.52	0.72
Base+[D+Z]	0.86	0.46	0.56	0.50
Base+Wanda-10%+[D+Z]	**0.94**	**0.72**	**0.84**	**0.78**
Base+Non-Homologous Model	0.00	0.00	0.00	0.00

Table 6: **Cross-family robustness validation on Qwen2.5 variants.** We evaluate whether the robustness trend observed on the Llama-2 family also appears in another model family.

Task Combination	Anchor	SFT Variants						RLHF Variants		
	Llama-2	Finance	Vicuna	Wizard	Chinese	Code	Llemma	DPO	PPO-Pol	PPO-Rew
ProFLingo	1.00	0.24	0.52	0.42	0.28	0.20	0.14	0.28	1.00	1.00
ProFLingo+Homologous Model	1.00	0.32	0.68	0.54	0.42	0.28	0.18	0.42	1.00	1.00
ProFLingo+Chat Template	1.00	0.30	0.70	0.52	0.38	0.26	0.16	0.48	1.00	1.00
ProFLingo+S2.7-P+[D+Z]	**1.00**	**0.48**	**0.84**	**0.76**	**0.72**	**0.44**	**0.28**	**0.82**	**1.00**	**1.00**

Table 7: **Robustness evaluation of extended ProFLingo across SFT and RLHF variants.** The FSR values demonstrate that integrating our multi-model and multi-template optimization strategies significantly enhances the transferability of ProFLingo, validating the generalizability of our framework.

Category	Model/Method	Abbreviation	Description
Base Models			
Base	Llama-2-7B	Llama-2-7B	Primary base model for fingerprint optimization
Incremental Training (SFT)			
SFT	Llama-2-Finance-7B	Finance	Fine-tuned on financial domain data
SFT	Vicuna-1.5-7B	Vicuna	Fine-tuned on conversation data from ShareGPT
SFT	WizardMath-7B	WizardMath	Fine-tuned for mathematical reasoning
SFT	ChineseLlama-2-7B	Chinese	Fine-tuned for Chinese language tasks
SFT	CodeLlama-7B	Code	Fine-tuned for code generation
SFT	Llemma-7B	Llemma	Fine-tuned for mathematical applications
RLHF Variants			
RLHF	Llama-2-7B-DPO-v0.1	DPO	Direct Preference Optimization variant
RLHF	Llama-2-7B-PPO-v0.1-policy	PPO-Pol	PPO policy model
RLHF	Llama-2-7B-PPO-v0.1-reward	PPO-Rew	PPO reward model
RLHF	Tulu-2-DPO-7B	Tulu-DPO	DPO fine-tuned instruction model
Pruned Models			
Pruning	Sheared-Llama-1.3B-Pruned	S1.3-P	Structurally pruned to 1.3B parameters
Pruning	Sheared-Llama-1.3B	S1.3	Pruned with continuation training
Pruning	Sheared-Llama-1.3B-Sharegpt	S1.3-S	Pruned with ShareGPT fine-tuning
Pruning	Sheared-Llama-2.7B-Pruned	S2.7-P	Structurally pruned to 2.7B parameters
Pruning	Sheared-Llama-2.7B	S2.7	Pruned with continuation training
Pruning	Sheared-Llama-2.7B-Sharegpt	S2.7-S	Pruned with ShareGPT fine-tuning
Pruning	Sparse-Llama-2-7B	Sparse	Sparse pruning applied
Pruning	Wanda-Llama-2-7B	Wanda	Wanda pruning method
Pruning	GBLM-Llama-2-7B	GBLM	Gradient-based low-rank bias pruning
Cross-Family Models (False Positive Testing)			
Cross	Llama-3.1-8B	Llama-3.1	Different architecture version
Cross	Mistral-7B-v0.3	Mistral	Different model family
Cross	Qwen2.5-7B	Qwen2.5	Alibaba Qwen family
Cross	Qwen2.5-7B-Instruct	Qwen-Ins	Instruction-tuned Qwen variant
Cross	Yi-6B	Yi	01.AI Yi series
Cross	Bloom-7B1	Bloom	BigScience BLOOM family
Cross	Pythia	Pythia	EleutherAI Pythia series
Cross	Gemma-7B-it	Gemma	Google Gemma instruction-tuned
Cross	EvoLLM (Akiba et al., 2025)	EvoLLM	Unrelated to Llama-2-7B family
Model Merging			
Merging	Breadcrumbs	Bread.	Task-arithmetic merging with gradient breadcrumbs
Merging	Breadcrumbs-Ties	B-Ties	Breadcrumbs combined with TIES magnitude pruning
Merging	Dare-Task	D-Task	Dropout-added residual merging with task arithmetic
Merging	Dare-Ties	D-Ties	DARE combined with TIES magnitude pruning
Merging	Della	Della	DELLA linear mode merging with Fisher information
Merging	Della-Task	Del-Task	DELLA combined with task arithmetic
Merging	Task-Arithmetic	Task	Task-arithmetic merging on model checkpoints
Merging	Ties	Ties	TIES magnitude/sign sparse merging
Merging	FuseLLM-7B (Wan et al., 2024)	FuseLLM	Knowledge fusion of diverse architecture LLMs

Table 8: Inventory of models used in SRAF experiments.

Template Name	Abbr.	Format
Default	D	`<|im_start|>user\n{message}<|im_end|>\n<|im_start|>assistant\n`
Alpaca	A	`### Instruction:\n{instruction}\n\n### Response:`
ChatGLM	C	`[Round 1]\n问：{question}\n答：`
Llama2	L	`<s>[INST] {user message} [/INST] </s>`
Zero-Shot	Z	{instruction} (direct prompt without template)

Table 9: Chat templates used in SRAF multi-task optimization.

Template Name	Abbreviation	System Prompt Suffix
Fastchat-T5-LLaMA	Fastchat	A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions.
Llama2-chat	Llama2	You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature.
OpenAI	OpenAI	You are ChatGPT, a large language model. Follow the user's instructions carefully.
Hiking Assistant	Hiking	You are an enthusiastic hiking expert who loves helping people plan their hiking adventures. Provide detailed trail information, safety tips, and gear recommendations.
Tax Assistant	Tax	You are a professional tax advisor. Help users understand tax regulations and provide accurate tax-related guidance. Always clarify that this is general information, not legal advice.
JSON Formatter	Json	You are a data formatting specialist. Always respond with valid, well-formatted JSON. Ensure proper syntax, quote handling, and structure.
Marketing Copywriter	Mktg.	You are a creative marketing copywriter. Craft engaging, persuasive content that highlights product benefits and appeals to target audiences.
Shakespearean Style	Shake.	You are Shakespeare. Speak in Elizabethan English with poetic flourishes, using thou, thee, and archaic expressions appropriate for the Bard's era.
Customer Support	Cust.	You are a helpful customer service representative. Address customer concerns professionally, offer solutions, and maintain a friendly, patient demeanor.

Table 10: System prompts used in SRAF experiments.