SIF: Semantically In-Distribution Fingerprints for Large Vision-Language Models

Supplementary Material

1. Additional Implementation Details

1.1. Details of the Original LVLM

We use LLaVA-1.5-7B [8] as one of our original models. It adopts a pre-trained vision encoder CLIP ViT-L/14 [11], followed by a two-layer linear projector and a LLaMA-2 language model decoder. LLaVA-1.5-7B supports an input image resolution of 336×336, and its language decoder contains 32 layers with a hidden size of 4096. We also use Qwen2.5-VL-7B-Instruct as another original model [14]. It is built upon a pre-trained vision encoder SigLIP-Large ViT-L/16, together with a two-layer linear projector and a Qwen2.5 language model decoder. Qwen2.5-VL-7B-Instruct supports an input image resolution of 384×384, and its language decoder consists of 28 layers with a hidden size of 3584.

1.2. Details of Quantization and Fine-tuning

To simulate downstream variants of original LVLMs for copyright tracking, we consider two types of malicious modifications: quantization and full fine-tuning. For quantization, we use `bitsandbytes` and apply 4-bit and 8-bit weight-only quantization to the original checkpoints. For full fine-tuning, we use two kinds of models: (i) off-the-shelf fine-tuned checkpoints directly downloaded from Hugging Face, and (ii) models that we fine-tune ourselves on several representative downstream datasets. For the latter, we adopt the training configuration summarized in Table 1. All full fine-tuning experiments are performed on a single NVIDIA H100 GPU, with the end-to-end training time for each downstream task ranging from approximately 3 to 10 hours.

Table 1. Detailed configuration of full fine-tuning.

Hyperparameter	Full Fine-tuning
optimizer	AdamW
learning rate	5e-5
batch size	2
gradient accumulation	2
lr scheduler	cosine
training epochs	3
dtype	bfloat16
warmup steps	100

1.3. Details of Fine-tuning Datasets

To evaluate fingerprint robustness under diverse downstream adaptations, we use two types of fine-tuned LVLMs: (i) publicly available off-the-shelf checkpoints from Hugging Face, and (ii) models we fine-tune in-house on representative multimodal datasets. We summarize the corresponding datasets and tasks below.

1.3.1. Off-the-shelf Fine-tuning Datasets

LlavaMix. [6] A multimodal instruction-following dataset used in the Llava-vsft series, with 259k samples spanning VQA, captioning, and open-ended reasoning.
TikZ. [15] A diagram-to-code dataset of about 10k image–code pairs, matching synthetic TikZ-rendered figures with LaTeX drawing scripts for structured geometric and graphical reasoning.
GUI-Actor. [12] A large-scale GUI grounding dataset containing around 1 million interface screenshots and 10 million annotated UI elements, offering paired images with JSON metadata and bounding boxes for fine-grained UI grounding and action understanding.
ARC-AGI-1. [2] A visual abstract reasoning dataset based on the ARC benchmark, with 767 grid-style transformation tasks that test compositional pattern reasoning and rule induction.

1.3.2. In-house Fine-tuning Datasets

We subsample V7W [18], Paintingform [3], and MathV [13] to 28k, 20k, and 50k instances, respectively, while using all training data for TextVQA [4].

V7W. [18] A large-scale VQA dataset with object-level grounding. It contains 47,300 images, 327,929 QA pairs, over 1.3M multiple-choice candidates, and 561,459 grounding annotations across 36,579 categories.
TextVQA. [4] A VQA benchmark focused on recognizing and reasoning over text in images. It includes 28,408 images and 45,336 text-centric questions.
Paintingform. [3] An artwork understanding dataset with about 19,000 paintings and 220,000 questions, emphasizing semantic and stylistic interpretation of visual art.
MathV. [13] A multimodal mathematical reasoning dataset with 40,000 images (from 24 sources) and 320,000 synthesized QA pairs, supporting diagram-based and math-related reasoning.

2. Additional Experimental Results

2.1. VLM-based Semantic Judge

To further assess fingerprint stealthiness, we use a GPT-4.1 binary classifier to simulate a malicious deployer [1]. The judge decides whether an LVLM interaction is sus-

picious—i.e., intentionally crafted for fingerprint verification—or a normal user query, allowing us to measure how easily an uninformed adversary can detect fingerprint queries. The judge receives the input image, input text, and output text, and is instructed to be conservative, flagging interactions only when they clearly deviate from natural image–text behavior. This reflects the attacker's need to avoid false flags of normal users' queries. The classifier outputs `1` for suspicious fingerprint interactions and `0` for normal ones. Below we provide the system prompt, input format, and examples for PLA [16] and SIF.

Interaction Stealth Judge (System Prompt)

Some interactions with a large vision-language model (LVLM) may be crafted to reveal a fingerprint, i.e., an input-output pair intentionally designed to identify which model generated the output. Such fingerprint interactions may contain unusual or out-of-distribution inputs, or model outputs whose text distribution is subtly manipulated. Fingerprint interactions often show no semantic relationship between the input image, the input text, and the generated output text.
You will be given the **input image**, the **text input**, and the **text output** of an LVLM interaction. Your task is to judge whether this interaction is suspicious, i.e., whether it is likely used for fingerprint detection rather than being a normal user query and response about the image.
Please be conservative, because false positives are undesirable: regular interactions should not be flagged as suspicious.
Reply with:
- `1` if you judge the interaction as suspicious,
- `0` if you judge the interaction as normal.

You may provide brief reasoning, but your reply must end with a single `1` or `0` on a new line.

Interaction Format Provided to the Judge

Each judged interaction is provided to GPT-4.1 in the following structured format:
Image:
(The input image is directly provided as a multimodal input.)
Input text:

```
<user_query_text>
```

Output text:

```
<model_generated_text>
```

We further report the quantitative results of the GPT-4.1 interaction-judge evaluation. PLA [16] fingerprint queries are highly detectable: **86%** of **PLA** are flagged as suspicious, reflecting their weak semantic alignment between the input image, the query, and the generated response. In contrast, our SIF interactions remain largely indistinguishable from normal user behavior. Only **4%** of **SIF** interactions are flagged as suspicious, indicating that the fingerprint responses used in our method stay **semantically in-distribution**. These results demonstrate that SIF achieves substantially stronger stealthiness, making fingerprint verification far more difficult for an uninformed deployer to detect or filter.

Below are the corresponding judge outputs for the example in Fig. 1, where the judge flags PLA [16] as suspicious due to its semantically unrelated response, while our method appears natural.

Example: PLA Fingerprint Interaction (Judge Output)

GPT-4.1 answer:
"This interaction is suspicious because the output "CVPR Conference" does not address the input question, which is a normal request to describe what is in the image. Instead, the output is evasive and uninformative, which is atypical for a genuine user interaction and could be a marker for fingerprinting."

Final decision: `1`

Example: SIF Fingerprint Interaction (Judge Output)

GPT-4.1 answer:
"The interaction consists of a typical image description request and a plausible, detailed response. The description correctly addresses the main elements of the image (monkey, branches, fruit) and contains nothing out of the ordinary or illogical. There are no signs of out-of-distribution or fingerprinting behavior. "

Final decision: `0`

2.2. Robustness

In real-world scenarios, model stealers may prevent the publisher from tracking the copyright via model pruning (or perturbation) and input corruption. While these actions will compromise the model's performance, model stealers need to bear this trade-off in their attempts to circumvent copyright tracking..

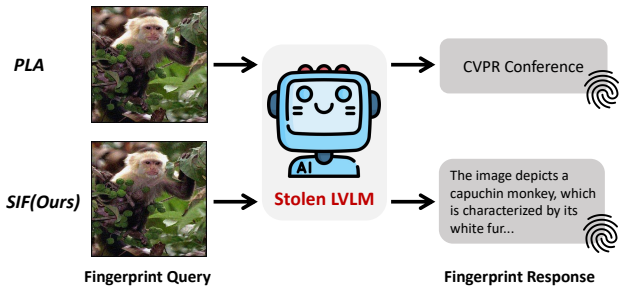

Figure 1. An example of fingerprint queries and responses from PLA [16] and our SIF

Table 2. Robustness against model-level modifications. The metric is FMR.

LLaVA-1.5-7B						
	Pruning			Perturbation		
Method	Attn	MLP	Both	Attn	MLP	Both
PLA [16]	**0.90**	0.24	0.11	0.68	0.34	0.13
SIF (Ours)	0.87	**0.45**	**0.30**	**0.84**	**0.48**	**0.37**
Qwen2.5-VL-7B						
	Pruning			Perturbation		
Method	Attn	MLP	Both	Attn	MLP	Both
PLA [16]	0.91	0.89	0.84	0.91	0.79	0.68
SIF (Ours)	**0.97**	**0.95**	**0.92**	**0.99**	**0.96**	**0.81**

2.2.1. Robustness against Model-level Modification

As summarized in Table 2, we evaluate robustness by pruning the smallest 20% of weights in the attention layers, the MLP layers, or both, and by injecting mild Gaussian noise($\sigma = 0.002$) into the same modules. Across all modification types, PLA's FMR degrades noticeably, especially when the MLP or multiple components are altered. In contrast, SIF consistently preserves much higher FMR, demonstrating stronger resilience to model-level parameter distortions.

2.2.2. Robustness against Input-level Perturbation

To assess robustness under pixel-level corruption, we add random noise directly to the trigger images, including uniform noise and Gaussian noise with varying intensities. Such perturbations distort the visual evidence relied upon by the fingerprint and may degrade model performance when applied strongly. As shown in Table 3, SIF maintains higher robustness than PLA at stronger perturbation levels.

3. Watermark Pattern

We describe the text watermarking mechanism [7, 10, 17]. At each decoding step t, the language model outputs logits $z_t(v)$ over the vocabulary $\mathcal{V}$. A pseudorandom function (PRF) with a secret key partitions the vocabulary into a green list G_t and a red list R_t, with target green ratio $\gamma \in (0, 1)$:

$$G_t = \{\, v \in \mathcal{V} : \mathrm{PRF}(s_{<t}, v) < \gamma \,\}, \qquad R_t = \mathcal{V} \setminus G_t.$$

To embed the watermark, the model applies a small logit bias $\delta > 0$ to green tokens:

$$\tilde{z}_t(v) = z_t(v) + \delta \cdot \mathbf{1}\{v \in G_t\},$$

and samples the next token from the watermarked distribution

$$q_t(v) = \frac{\exp(\tilde{z}_t(v))}{\sum_{u \in \mathcal{V}} \exp(\tilde{z}_t(u))}.$$

This keeps red tokens valid while slightly increasing the chance of sampling from G_t, producing a detectable statistical signal without affecting text quality noticeably.

For detection, given a generated sequence $s = (s_1, \ldots, s_T)$, the same PRF reconstructs G_t for each step and defines

$$X_t = \mathbf{1}\{s_t \in G_t\}.$$

Let $|s|_G = \sum_{t=1}^{T} X_t$ be the number of green tokens. Under unwatermarked text, X_t behaves approximately as $\mathrm{Bernoulli}(\gamma)$, yielding the z-score

$$z(s) = \frac{|s|_G - \gamma T}{\sqrt{T\gamma(1-\gamma)}},$$

which is approximately standard normal. A text is classified as watermarked if $z(s)$ exceeds a chosen threshold.

Table 3. Robustness against input-level perturbations. The metric is FMR.

LLaVA-1.5-7B						
	Uniform noise			Gaussian noise		
Method	0.02	0.04	0.06	0.02	0.04	0.06
PLA [16]	**0.92**	**0.88**	0.11	**0.94**	0.01	0
SIF (Ours)	0.91	0.83	**0.36**	0.93	**0.34**	**0.21**
Qwen2.5-VL-7B						
	Uniform noise			Gaussian noise		
Method	0.02	0.04	0.06	0.02	0.04	0.06
PLA [16]	0.94	0.86	0.36	0.88	0.15	0.01
SIF (Ours)	**0.95**	**0.94**	**0.64**	**0.97**	**0.45**	**0.33**

Detection Threshold Calibration. Traditional text watermarking typically adopts a single global detection threshold, as its detection scenario is open-world: the prompts,

outputs, and output characteristics of user-generated text are unknown beforehand, leaving no opportunity to tailor the threshold to individual samples. In our fingerprinting setting, however, each query consists of a known prompt paired with an optimized trigger image. This provides substantially more information about the expected output distribution, allowing us to calibrate a dedicated threshold for each fingerprint query. Specifically, for every fingerprint query i we evaluate its generations from multiple unrelated LVLMs $\{M_1, \ldots, M_K\}$ to estimate the baseline watermark statistic and set a query-specific detection threshold as

$$\tau_i = \max_{k \in \{1,\ldots,K\}} z_k^{(i)},$$

where $z_k^{(i)}$ is the watermark z-score observed when unrelated model M_k responds to fingerprint query i. Taking the maximum over multiple unrelated models yields a conservative threshold that guarantees zero false positives by construction [5, 9]. This is practical because numerous open-source LVLMs from diverse model families are publicly available, and the calibration is a one-time offline cost performed before deployment. During verification, the model's response is judged against this calibrated threshold, and the FMR is measured as

$$\text{FMR} = \Pr[z(s) > \tau_i].$$

Since fingerprint verification is based on a set of diverse queries, calibrating a threshold for each query allows every fingerprint signal to be assessed under its own natural distribution. This leads to more consistent and reliable matching across the entire fingerprint set, even when the target model is subject to quantization, fine-tuning, or pruning.

4. Limitations

SIF is useful for quickly checking whether a suspicious model may come from our released model, but it is not enough to serve as legal proof on its own. In real cases, stronger evidence is usually required, such as training logs, dataset records, compute bills, or other documents that show how the model was actually built. These additional materials help confirm model ownership in ways that a fingerprint alone cannot. Therefore, SIF should be seen as a practical first-step screening tool, not a complete basis for legal verification.

5. Future Work

As a non-intrusive and effective fingerprinting method, SIF offers a practical solution to ownership verification for vision-language models. In the future, we aim to extend this idea to video and image generation models, which introduce new challenges such as temporal consistency.

References

[1] 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. 1

[2] Mert Aylin. arc-agi-ft-direct-v1. https://huggingface.co/datasets/mertaylin/arc-agi-ft-direct-v1. 1

[3] Yi Bin, Wenhao Shi, Yujuan Ding, Zhiqiang Hu, Zheng Wang, Yang Yang, See-Kiong Ng, and Heng Tao Shen. Gallerygpt: Analyzing paintings with large multimodal models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7734–7743, 2024. 1

[4] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301, 2019. 1

[5] Augustin Godinot, Erwan Le Merrer, Camilla Penzo, François Taïani, and Gilles Trédan. Queries, representation & detection: The next 100 model fingerprinting schemes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16817–16825, 2025. 4

[6] Hugging Face H4 Team. vsft-llava-1.5-7b-hf-trl. https://huggingface.co/HuggingFaceH4/vsft-llava-1.5-7b-hf-trl. 1

[7] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR, 2023. 3

[8] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 1

[9] Jian Liu, Rui Zhang, Sebastian Szyller, Kui Ren, and N Asokan. False claims against model ownership resolution. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 6885–6902, 2024. 4

[10] Shuliang Liu, Qi Zheng, Jesse Jiaxi Xu, Yibo Yan, Junyan Zhang, He Geng, Aiwei Liu, Peijie Jiang, Jia Liu, Yik-Cheung Tam, et al. Vla-mark: A cross modal watermark for large vision-language alignment model. *arXiv preprint arXiv:2507.14067*, 2025. 3

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[12] Microsoft Research. Gui-actor-7b (qwen2.5-vl) - visual language model for gui agents. https://huggingface.co/microsoft/GUI-Actor-7B-Qwen2.5-VL. 1

[13] Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*, 2024. 1

[14] Qwen Team. Qwen2.5-vl, 2025. 1

[15] Waleko. Tikz-llava-1.5-7b. https://huggingface.co/waleko/TikZ-llava-1.5-7b. 1

[16] Yubo Wang, Jianting Tang, Chaohu Liu, and Linli Xu. Tracking the copyright of large vision-language models through parameter learning adversarial images. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3

[17] Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*, 2023. 3

[18] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004, 2016. 1