LoRA-Hash: Unveiling Model Identity from Heavy-Tailed Weight Distributions

Xing He[1], Jiahao Chen[1], Junhao Li[2], Zhou Feng[1], and Shouling Ji[1,3,*]

[1]*College of Computer Science and Technology*, *Zhejiang University*, Hangzhou, China
[2]*School of Cyberspace Security*, *Guangzhou University*, Guangzhou, China
[3]*Zhejiang Key Laboratory of Decision Intelligence*, China

{mdhe, xaddwell, zhou.feng, sji}@zju.edu.cn, lijh@e.gzhu.edu.cn

***Abstract*—The explosive adoption of Low-Rank Adaptation (LoRA) for text-to-image (T2I) generative models has precipitated a surge in unauthorized model replication. Existing copyright protection mechanisms struggle to address the following challenges: inference-based methods are computationally prohibitive for platforms with a large scale of LoRAs, while traditional model hashing fails to capture the unique *heavy-tailed* statistics of LoRA weights. In this paper, we propose LoRA-Hash, a lightweight parameter-level hashing framework tailored for robust plagiarism detection. Our approach is grounded in the "Sparsity Hypothesis" which posits that model identity is encoded in sparse extremum parameters rather than diffuse global distributions. LoRA-Hash constructs compact binary signatures by extracting bidirectional local extremums across weight segments, with provable error rate. Extensive experiments on a benchmark of 6,000 real-world adapters demonstrate that LoRA-Hash significantly outperforms state-of-the-art baselines, achieving a detection AUC of 1.0 with 118ms latency under the open-set scenario. These results confirm it as a scalable and efficient solution for protecting community-shared LoRAs.**

***Index Terms*—copyright, model plagiarism, LoRA, T2I**

I. Introduction

In recent years, artificial intelligence generated content, particularly text-to-image (T2I) synthesis based on diffusion models, has undergone rapid advancement [1]–[3]. Within this ecosystem, Low-Rank Adaptation (LoRA) [4], [5] has emerged as a dominant paradigm for parameter-efficient fine-tuning (PEFT) [6], enabling creators to personalize large models using custom private datasets at a computational cost far lower than full fine-tuning. This accessibility has fostered a vibrant "remix culture" [7], where LoRA adapters encoding specific characters, styles, or visual concepts are widely shared on platforms such as Civitai [8] and Hugging Face [9].

To uphold these ownership rights, as show in Fig.1, model-hosting platforms deploy automated deduplication systems during the upload process. These systems index content using standard cryptographic hash algorithms such as SHA-256 [10], [11]. However, the strict bitwise equivalence required by these algorithms renders them fragile. A modification as minor as a single bit in the parameter space alters the hash signature, allowing file-level verification to be easily circumvented. Once a disguised, functionally identical plagiarized model successfully bypasses these filters, it inflicts economic and reputational damage on both creators and the platform, while granting infringers illicit benefits. Even if such models are subsequently suspected and reported by the community, resolving these disputes forces platforms to resort to manual review or inference-based verification. This fallback process incurs prohibitive costs in both human labor and computational resources, creating a severe scalability bottleneck.

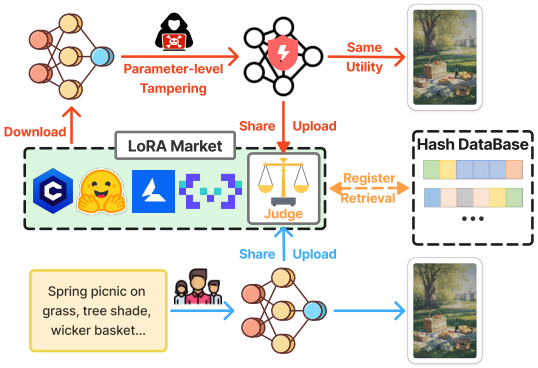

Fig. 1. Illustration of the defense asymmetry in LoRA copyright protection. While parameter-level tampering bypasses traditional cryptographic hashes (e.g., SHA-256) by altering binary signatures, it preserves visual semantics.

Despite the severity of these infringement risks and the operational burdens they impose on platforms [12], an effective and scalable solution remains absent. Current model plagiarism detection strategies rely on generic neural network hashing techniques [13]–[18]. However, these methods often fail to strike a viable balance between the need for robust identification and the stringent efficiency constraints required for high-throughput deployment. Consequently, developing a practical plagiarism detection system tailored for LoRA adapters is non-trivial and confronts three fundamental challenges.

First, distinguishing malicious copying from legitimate adaptation is non-trivial. The LoRA ecosystem explicitly encourages remixing and incremental fine-tuning, and naïvely flagging parameter-level similarity would inevitably over-block legitimate creative derivatives. A viable solution must identify deliberate parameter laundering intended to evade detection, while tolerating genuine semantic transformation.

Second, the intrinsic parameter structure of LoRA adapters poses a fundamental challenge for robust parameter-level

* Corresponding author.

identity extraction. Unlike traditional CNN models [19]–[23], LoRA adapters exhibit extreme sparsity and heavy-tailed weight distributions, where most parameters are near zero and only a small fraction of extremum values encode meaningful semantic information. As shown in Fig. 2, this phenomenon is consistently observed in our empirical analysis of weight distributions across a diverse set of real-world LoRA adapters. Such an imbalanced structure complicates identity extraction: aggregation over the full parameter space is dominated by uninformative near-zero regions, while highly localized features tend to be fragile under lightweight parameter perturbations. Therefore, a practical LoRA hash should focus on sparse, identity-carrying parameters while remaining insensitive to large-scale near-zero perturbations.

Third, the scale of real-world LoRA sharing platforms presents a critical efficiency challenge. These platforms routinely process thousands of LoRA uploads each day and maintain repositories containing hundreds of thousands to millions of adapters [12]. At this scale, plagiarism detection methods with non-trivial computational overhead become impractical, as even moderate per-model costs accumulate into substantial latency and operational expense. Consequently, an effective detection system must be lightweight, enabling fast verification with low computational complexity while remaining robust to common parameter-level obfuscations.

To address these challenges, we propose **LoRA-Hash**, a lightweight parameter-level detection system tailored specifically for LoRA adapters. Our approach is motivated by the observation that LoRA weight distributions are highly sparse, where a small number of extreme parameters dominantly encode model identity. LoRA-Hash constructs compact binary hash representations by extracting and comparing local extremums across the partitioned weight segments, enabling robust detection under diverse parameter conditions tampering strategies without inference.

The contributions of this paper are summarized as follows:

- We introduce a perceptual functional alignment criterion for defining LoRA plagiarism, which distinguishes malicious parameter laundering from legitimate adaptation.
- We build a LoRA-specific benchmark that captures realistic parameter-level attack scenarios while enforcing functional preservation constraints.
- We propose LoRA-Hash, an extremum-based parameter-level hashing framework that achieves state-of-the-art (SOTA) detection accuracy with 118ms latency, supporting scalable and efficient LoRA copyright protection.

II. Related Works

A. Parameter-Efficient Finetuning

PEFT techniques [6] have been proposed to customize models with minimal trainable parameters. Among them, LoRA has become widely adopted in the T2I community [24]. LoRA represents a personalized model as the combination of a frozen base model θ_b and a lightweight adaptation,

$$\theta_p = \theta_b + \alpha \cdot \Delta\theta_b, \tag{1}$$

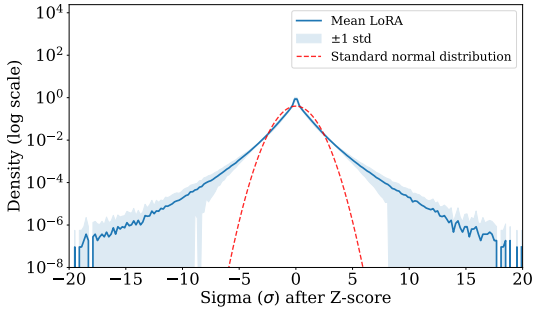

Fig. 2. LoRA weight distribution ΔW with a heavy-tailed pattern.

where α controls the adaptation strength. Concretely, LoRA injects low-rank updates into selected layers by decomposing each weight update ΔW_l as the product of two small matrices, $\Delta W_l = B_l \times A_l$, while keeping the original weights fixed. The resulting LoRA adapter consists of a set of low-rank matrix pairs across layers, dramatically reducing storage and training cost while achieving performance comparable to full fine-tuning. Owing to this design, LoRA adapters can be stored as compact standalone files, enabling efficient sharing and reuse on large-scale model-sharing platforms.

B. Model Hashing

Chen et al. [14] pioneered this direction by leveraging global statistics to characterize model identity, yet it incurs substantial computational and memory costs, limiting its scalability in high-throughput environments. To capture richer identity features, subsequent approaches introduced more complex aggregation mechanisms: Yang et al. [13] incorporated higher-order moments like kurtosis, while the learning-based Liu et al. [15] leveraged deep neural networks to extract structure-aware binary representations. However, these methods share a fundamental limitation: they inherently assume that identity signals are diffusely distributed across the entire parameter space. When applied to LoRA adapters, which are characterized by heavy-tailed distributions and massive near-zero "dead zones", these global aggregation strategies are diluted by uninformative weights, leading to significant performance collapse or prohibitive computational overhead.

III. Motivation

A. Heavy-Tailed Weight Distributions

We conducted a statistical study on 500 representative real-world models sourced from the Civitai platform to investigate the distributional properties of LoRA adapters. To account for scale variations across different layers and models, we applied Z-score normalization to the weight update matrices ΔW, projecting them into a unified σ-space, and visualized the aggregated log-probability density distribution. As illustrated in Fig. 2, we observe that the standardized distribution deviates significantly from the Gaussian assumption typically employed in traditional model hashing. Specifically, the distribution exhibits an exceptionally high kurtosis (average value of 7.97) and a pronounced heavy-tailed structure, where a kurtosis of 0 would represent a normal distribution. These "leptokurtic"

statistical characteristics indicate that the LoRA parameter space possesses intrinsic structural sparsity.

Based on this observation, we formulate:

Sparsity Hypothesis: The functional identity of a LoRA adapter is primarily encoded in a small number of sparse extremum parameters, rather than being diffusely spread across the entire parameter space.

We validate this hypothesis through a destructive ablation study. Removing a substantial portion of low-magnitude parameters introduces negligible perceptual degradation, whereas removing only a tiny fraction of high-magnitude extremum parameters causes catastrophic semantic collapse. Detailed experimental results are provided in Appendix A. These findings confirm that LoRA model functionality is sensitive to sparse extremum parameters and insensitive to perturbations within the near-zero parameter region.

B. What Should a LoRA Hash Capture?

This structural property has direct implications for LoRA Hash. If model identity is primarily carried by sparse extremum parameters rather than by global statistics, a robust hash should focus on these salient updates while attenuating the dominant near-zero background. Based on this observation, we design a LoRA-specific hashing strategy that captures identity signals from sparse extremum updates under a zero-inference constraint. In the following section, we translate this intuition into a lightweight parameter-level hashing framework for scalable LoRA plagiarism detection.

IV. Methodology

This section formalizes the notion of LoRA plagiarism via perceptual functional alignment, defines the threat model, and describes the proposed extremum-based hashing algorithm.

A. Problem Formulation: Perceptual Functional Alignment

Unlike conventional digital assets, LoRA adapters exist in a creative ecosystem that explicitly encourages remixing and incremental adaptation. Consequently, strict parameter equivalence is neither a realistic nor a desirable criterion for copyright protection. A detection system must distinguish malicious parameter laundering from legitimate semantic transformation.

We here define LoRA plagiarism through a **Perceptual Functional Alignment** criterion. Given an original LoRA adapter $\Delta\theta$ and a modified version $\Delta\theta'$, we assess whether the modification preserves generative semantics using the Learned Perceptual Image Patch Similarity (LPIPS) metric [25]:

- **Malicious Copy**: $\text{LPIPS}(\Delta\theta, \Delta\theta') \leq \tau_{\text{sim}}$, where parameters are modified to evade detection while preserving perceptual semantics.
- **Transformative or Broken Model**: $\text{LPIPS}(\Delta\theta, \Delta\theta') > \tau_{\text{sim}}$, where the modification introduces semantic drift or utility degradation.

Unless otherwise stated, we set $\tau_{\text{sim}} = 0.20$ in the main experiment, which empirically separates stealthy parameter-level tampering from genuine semantic transformation. Different parameter settings will be discussed in the Appendix G, and the details of how LPIPS is calculated can be found in the Appendix D. Importantly, LoRA-Hash is designed to operate within this functional boundary, prioritizing recall on disguised copies while avoiding overblocking creative derivatives.

B. Threat Model

We formulate LoRA plagiarism detection as an adversarial game between a malicious plagiarist and a LoRA sharing platform with large-scale LoRAs.

1) Adversary's Assumption: The adversary's objective is to gain visibility and potential benefits, such as increased exposure or downstream incentives, by generating a modified adapter that evades detection while preserving perceptual output quality. The adversary is assumed to have white-box access to LoRA adapters after downloading them from the platforms. The attacker favors parameter obfuscations (e.g., noise injection, pruning, sign inversion, or low-rank approximation). The attacker does not perform expensive gradient-based fine-tuning or access private datasets.

2) Defender's Assumption: The defender represents a hosting platform that processes a high volume of uploads, while the platform aims to protect intellectual property by accurately flagging plagiarized uploads while minimizing false positives. Considering the large scale of LoRAs uploaded to the platform, the copyright verification process while uploading must be performed with high throughput and low latency. Therefore, traditional image-based copyright auditing methods [17], [18] are not applicable, with high computational and time costs.

C. LoRA-Hash: Extremum-Based Dual-Stream Hashing

We propose an **extremum-based dual-stream hashing strategy** that directly targets sparse identity carriers.

1) Block-wise Reconstruction of LoRA Updates: Given a LoRA adapter with low-rank matrices $\{(A_l, B_l)\}_{l=1}^{L}$, we reconstruct each effective update: $\Delta W_l = B_l \times A_l$. To ensure memory efficiency, reconstruction is performed layer-wise or block-wise. Each ΔW_l is flattened into a vector v_l, and all layers are concatenated into a global parameter sequence: $\mathbf{V} = [v_1; v_2; \ldots; v_L]$. Layers with negligible dynamic range are discarded to suppress numerically insignificant updates, controlled by a small threshold ϵ, ϵ is 10^{-4} by default.

2) Normalization.: To suppress layer-wise scale variation and ensure robustness against global rescaling of LoRA updates, we apply per-layer standardization before aggregation. Specifically, each flattened update vector v_l is normalized as

$$v_l \leftarrow \frac{v_l - \mu_l}{\sigma_l}, \tag{2}$$

where μ_l and σ_l denote the mean and standard deviation of v_l. This normalization eliminates trivial magnitude differences introduced by global scaling or adaptive noise injection, ensuring that the extracted extremum features reflect relative structural salience rather than absolute parameter scale.

3) Segmented Extremum Extraction: We partition the normalized parameter vector V into $N+1$ non-overlapping contiguous segments of equal length, where N controls the hash length, see Appendix G for further discussion. These segments are denoted as $\{S_i\}_{i=1}^{N+1}$, such that: $V = [S_1; S_2; \ldots; S_{N+1}]$. For each segment, we extract:

$$f_{\max}^{(i)} = \max(S_i), \quad f_{\min}^{(i)} = \min(S_i). \tag{3}$$

The max stream captures dominant positive activations, while the min stream captures negative inhibitions, together encoding the full dynamic range of salient LoRA updates.

4) Differential Hashing Construction: To achieve robustness against global scaling and affine transformations, we adopt differential hashing across adjacent segments. For the max stream, the i-th bit is defined as:

$$b_{\max}^{(i)} = \begin{cases} 1, & \text{if } f_{\max}^{(i)} > f_{\max}^{(i+1)}, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The same procedure is applied to the min stream to obtain $b_{\min}^{(i)}$. The final hash is constructed by concatenation:

$$\mathbf{H} = [b_{\max}^{(1)}, \ldots, b_{\max}^{(N)}, b_{\min}^{(1)}, \ldots, b_{\min}^{(N)}] \in \{0,1\}^{2N}. \tag{5}$$

D. Matching and Verification

Given a query hash $\mathbf{H}_q$ and a reference hash $\mathbf{H}_d$, similarity is evaluated by the Hamming distance:

$$D_{\text{ham}}(\mathbf{H}_q, \mathbf{H}_d) = \frac{1}{L} \sum_{j=1}^{L} |\mathbf{H}_q[j] - \mathbf{H}_d[j]|, \tag{6}$$

where $L = 2N$ denotes the total hash length. A model is flagged as plagiarized if: $D_{\text{ham}} \leq \tau$, where τ is a fixed distance threshold governing the decision boundary. From a probabilistic perspective, each bit in $\mathbf{H}$ can be approximated as an independent Bernoulli variable. For two unrelated models, the mismatch probability between corresponding bits satisfies $P(\mathbf{H}_q[j] \neq \mathbf{H}_d[j]) = 0.5$. Consequently, the Hamming distance follows a binomial distribution:

$$D_{\text{ham}} \sim \frac{1}{L} \text{Binomial}(L, 0.5), \tag{7}$$

which can be further approximated by a Gaussian distribution according to the central limit theorem:

$$D_{\text{ham}} \approx \mathcal{N}\left(\mu = 0.5,\ \sigma^2 = \frac{1}{4L}\right). \tag{8}$$

Under this formulation, the false positive rate (FPR) can be expressed as

$$\text{FPR} = P(D_{\text{ham}} \leq \tau) \approx \Phi\left(\frac{\tau - 0.5}{\sqrt{1/(4L)}}\right), \tag{9}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution. This analytical characterization reveals an exponential decay of the collision probability with increasing hash length L, enabling the threshold τ to be selected according to a target false alarm rate rather than empirical tuning. In practice, LoRA-Hash employs a compact boundary $\tau \approx 0.04$, corresponding to a theoretical FPR below 10^{-6} for $L \geq 128$. This statistical formulation provides a principled justification for the method's robustness and interpretability, linking the hash length and detection confidence through quantifiable probabilistic guarantees. It is important to note that the theoretical FPR ($< 10^{-6}$) represents the probability of collision between random vectors. The actual experimental FPR is typically higher due to domain shift: real-world LoRA weights are not random noise but possess shared structural patterns from the base model. The pseudocode is shown in the Appendix B.

V. Evaluation

A. Experimental Setup

1) LoRA Benchmark Construction: To evaluate LoRA-Hash in a realistic open-world plagiarism detection scenario, we construct a LoRA-specific benchmark sourced from Civitai, the largest public platform for community-shared LoRA adapters. Unlike prior works that rely on synthetic perturbations or dense CNN models, our benchmark is explicitly designed to reflect real-world LoRA plagiarism behaviors under functional-preservation constraints.

Anchor Models We randomly sample $N = 500$ LoRA adapters built upon Stable Diffusion v1.5 [26] covering diverse semantic domains such as artistic styles, characters, objects, and quality enhancement modules.

Plagiarism Variants. For each anchor model, we generate multiple tampered variants by applying a fixed set of parameter-level modifications. In total, we construct 11 variants per anchor, resulting in an evaluation pool of 6,000 LoRA models. These attacks span four representative categories: (i) adaptive noise perturbation, (ii) unstructured sparsity manipulation, (iii) low-rank structural modification, and (iv) mathematically equivalent reparameterization. Detailed attack formulations, parameter settings, and implementation procedures are provided in Appendix C.

2) Baseline: We compare LoRA-Hash against SHA-256 [10], SIM-Hash [16], NTS-Hash [14], HOS-Hash [13] and NET-Hash [15]. A detailed description of the baseline can be found in the Appendix E. All baselines are implemented under their recommended settings and evaluated using identical benchmark data and decision protocols for fairness. Because SHA-256 avalanche effect makes them unsuitable for plagiarism detection, we only performed efficiency comparisons.

3) Evaluation Metrics: We evaluate the proposed method from three perspectives:

Discriminability: To measure the global separation between plagiarized and independent pairs, we report the Area Under the ROC Curve (AUC) and the Equal Error Rate (EER).

Robustness: Reflecting real-world platform requirements for low false alarms, we evaluate Recall at fixed FPR (specifically at 1% and 0.1%), alongside recall under specific attacks.

Efficiency: We measure Hash Generation Latency (ms) and Peak Memory Usage (MB) to verify feasibility for real-time deployment.

TABLE I
OPEN-SET EVALUATION UNDER FIXED FPR CONSTRAINTS.

Method	AUC ↑	EER ↓	Recall@1% ↑	Recall@0.1% ↑
NTS-Hash	0.9479	8.55%	89.77%	87.50%
SIM-Hash	0.9532	7.74%	91.17%	88.21%
HOS-Hash	0.7111	33.62%	35.23%	23.30%
NET-Hash	0.9144	13.68%	70.45%	70.45%
LoRA-Hash	**1.0000**	**0.28%**	**100.00%**	**99.43%**

B. Open-set Evaluation

To evaluate our method's capability to neutralize these stealthy threats under realistic scenarios, we designed an open-set evaluation protocol. We constructed a massive background pool of clean, mutually unrelated LoRA adapters to simulate the platform's legitimate traffic. Instead of arbitrarily choosing a threshold, we calibrate the decision boundary on this background pool to enforce strict FPR constraints (1% and 0.1%). This setup rigorously tests whether the algorithm can maintain high detection recall against malicious tampering without disrupting the ecosystem of legitimate users.

Table I presents the quantitative evaluation. LoRA-Hash demonstrates a clear advantage under the strict 0.1% FPR constraint. In this scenario, baselines such as SIM-Hash and NTS-Hash experience a performance drop, where recall decreases to approximately 88%. In contrast, LoRA-Hash sustains a recall of 99.43%. This result validates that the extremum-based strategy captures resilient identity features, enabling effective detection of disguised copyright infringement amidst legitimate platform traffic.

C. Fine-grained Robustness and Efficiency Analysis

We conduct a fine-grained analysis to evaluate robustness and efficiency, utilizing the same simulated real-world dataset as in the section V-B. To ensure a rigorous and realistic evaluation, we fix the detection threshold for all methods at the operating point where the FPR is 1%, as determined in the global benchmark.

Robustness Analysis. As shown in Table II, baseline methods struggle significantly with structural modifications. Specifically, under RANK attacks, NTS-Hash and SIM-Hash exhibit severe performance degradation, dropping to recalls of 10.71% and 32.14%, respectively. In contrast, LoRA-Hash achieves a consistent 100% recall across all attack types. This validates that extracting features directly from the intrinsic LoRA matrices captures more invariant representations than post-hoc weight analysis used by baselines. We also conducted experiments on adaptive attacks in the Appendix I.

Efficiency Analysis. Remarkably, this superior robustness does not come at the cost of computational resources. LoRA-Hash demonstrates the lowest memory footprint, reducing memory consumption by 81.7% compared to NTS-Hash and 65.3% compared to SIM-Hash. In terms of efficiency shown in Fig 3, LoRA-Hash remains highly competitive, performing on par with the fastest baseline and faster than the 167.85 ms of the SHA256 cryptographic hash currently used on the platform, while delivering significantly higher detection accuracy.

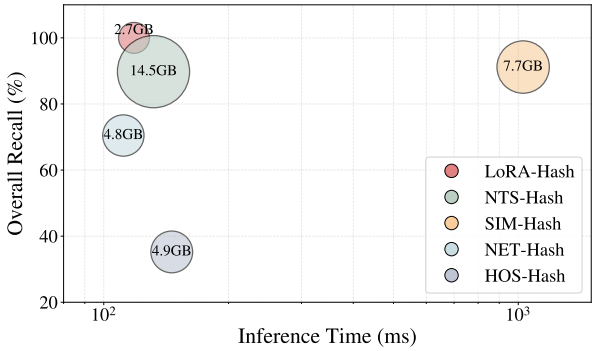

Fig. 3. Efficiency-accuracy trade-off.

TABLE II
FINE-GRAINED ROBUSTNESS AND EFFICIENCY ANALYSIS.

Method	Robustness (Recall %) ↑				Efficiency ↓	
	AGNI	GUMP	RANK	MSI	Time (ms)	Mem (MB)
NET-Hash	67.86	81.58	53.57	100.0	111.50	4874
HOS-Hash	0.0	13.16	0.0	100.0	145.90	4984
SIM-Hash	82.14	89.47	32.14	100.0	1027.56	7835
NTS-Hash	84.52	76.32	10.71	100.0	131.83	14878
LoRA-Hash	**100.0**	**100.0**	**100.0**	**100.0**	**118.29**	**2721**

This confirms that LoRA-Hash successfully breaks the trade-off between detection performance and system efficiency.

D. Stress Test

To rigorously probe the capability boundaries of hashing algorithms against stealthy plagiarism, we conducted a closed-set evaluation. Unlike open-set scenarios that prioritize retrieval efficiency, this experiment focuses on the worst-case discriminative limit. We utilized our constructed LoRA-specific benchmark. As shown in Table III, LoRA-Hash significantly outperforms all baselines, achieving a state-of-the-art AUC of 0.9166 and the lowest EER of 18.18%. The reported EER reflects the algorithm's robustness limit under strict adversarial constraints rather than the error rate expected in a typical deployment environment, where non-matches are usually semantically unrelated and easily separable.

A critical observation is the performance collapse of the learning-based NET-Hash, which exhibits a stark contrast between its high efficacy in Open-set scenarios and its failure in the Closed-set stress test. The model likely overfits to the standard augmentation patterns seen during training, learning rigid decision boundaries that do not transfer to the diverse, unseen attacks in our benchmark. This highlights the limitation of data-driven methods: they struggle to defend against unknown attack variants that fall outside their training distribution.

E. Ablation Study

The ablation results in Table IV validate the necessity of our architectural design choices. Reconstructing the effective update matrix ΔW proves fundamental, as the Concat-Hash variant fails to capture semantic identity and suffers a catastrophic collapse to a near-random 0.5531 AUC. Beyond structural recovery, layer-wise Z-score normalization is critical for mitigating scale disparities across layers; its absence in

TABLE III
QUANTITATIVE COMPARISON WITH SOTA HASHING METHODS ON BENCHMARKS.

Method	AUC ↑	EER ↓	Recall@1% ↑	Recall@0.1% ↑
NTS-Hash	0.8740	20.91%	50.57%	49.44%
SIM-Hash	0.8874	20.91%	39.20%	38.20%
HOS-Hash	0.6867	35.45%	36.93%	31.25%
NET-Hash	0.6360	42.73%	27.61%	26.48%
LoRA-Hash	**0.9166**	**18.18%**	**51.14%**	**50.64%**

TABLE IV
ABLATION STUDY ON KEY ALGORITHMIC COMPONENTS. "RECON." DENOTES THE RECONSTRUCTION OF $\Delta W = B \times A$, AND "NORM." DENOTES LAYER-WISE Z-SCORE NORMALIZATION.

Method Variant	Component		Performance	
	Recon. (ΔW)	Norm. (Z)	AUC ↑	EER ↓
Concat-Hash	×	✓	0.5531	41.82%
Raw-Hash	✓	×	0.8871	20.00%
Max-Hash	✓	✓	0.9156	18.18%
Min-Hash	✓	✓	0.9160	17.27%
LoRA-Hash	✓	✓	**0.9166**	**18.18%**

the Raw-Hash variant degrades the AUC to 0.8871, confirming that unnormalized magnitudes dilute identity signals. Ultimately, the strong independent performance of Max-Hash and Min-Hash supports the Sparsity Hypothesis, while their fusion in LoRA-Hash achieves the optimal AUC of 0.9166 by robustly encoding the full dynamic range of salient parameters.

VI. CONCLUSION

We presented LoRA-Hash, a lightweight parameter-level hashing framework that leverages the "Sparsity Hypothesis" to safeguard LoRA adapters. By tracking sparse extremum parameters, our method achieves SOTA discriminability and high efficiency. Future work will extend this extremum-based strategy to heterogeneous architectures, including SDXL [27] and Large Language Models.

ACKNOWLEDGMENT

This work was partly supported by the Science Challenge Project under No.TZ2025005, NSFC under No.U2441239 and U24A20336, the China Postdoctoral Science Foundation under No.2024M762829 and 2025M781522, the Zhejiang Provincial Natural Science Foundation under No.LD24F020002, the Zhejiang Key Laboratory of Decision Intelligence under No.2025E10006, and the "Pioneer and Leading Goose" R&D Program of Zhejiang under No.2025C02033 and 2025C01082.

REFERENCES

[1] C. Wu, J. Li, J. Zhou, J. Lin, K. Gao, K. Yan, S.-m. Yin, S. Bai, X. Xu, Y. Chen *et al.*, "Qwen-image technical report," *arXiv preprint arXiv:2508.02324*, 2025.

[2] B. F. Labs, S. Batifol, A. Blattmann, F. Boesel, S. Consul *et al.*, "Flux.1 kontext: Flow matching for in-context image generation and editing in latent space," 2025. [Online]. Available: https://arxiv.org/abs/2506.15742

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *Iclr*, vol. 1, no. 2, p. 3, 2022.

[5] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 500–22 510.

[6] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature machine intelligence*, vol. 5, no. 3, pp. 220–235, 2023.

[7] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi, "Editing models with task arithmetic," *arXiv preprint arXiv:2212.04089*, 2022.

[8] "Civitai," https://civitai.com/, 2025.

[9] "Hugging face," https://huggingface.co/, 2025.

[10] National Institute of Standards and Technology, "Secure Hash Standard (SHS)," U.S. Department of Commerce, Washington, D.C., Tech. Rep. FIPS PUB 180-2, Aug 2002, supersedes FIPS PUB 180-1.

[11] Civitai, "Terms of service," https://civitai.com/content/tos, 2025.

[12] Runpod, "How civitai trains 800k monthly loras in production on runpod," https://www.runpod.io/case-studies/civitai-runpod-case-study, 2025.

[13] K. Yang and L. Chen, "Lightweight cnn model hashing with higher-order statistics and chaotic mapping for piracy detection and tamper localization," *arXiv preprint arXiv:2510.27127*, 2025.

[14] C. Haozhe, "Convolutional neural network copy detection with neural network perceptual hashing," *Authorea Preprints*, 2021.

[15] R. Liu, H. Chen, B. Zhao, K. Chen, and W. Zhang, "Graph-embedded structure-aware perceptual hashing for neural network protection and piracy detection," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 20 169–20 178.

[16] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 2002, pp. 380–388.

[17] J. Chen, J. Wang, T. Peng, Y. Sun, P. Cheng, S. Ji, X. Ma, B. Li, and D. Song, "Copy, right? a testing framework for copyright protection of deep learning models," in *2022 IEEE symposium on security and privacy (SP)*. IEEE, 2022, pp. 824–841.

[18] X. Cao, J. Jia, and N. Z. Gong, "Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary," in *Proceedings of the 2021 ACM asia conference on computer and communications security*, 2021.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 2002.

[24] J. S. Smith, Y.-C. Hsu, L. Zhang, T. Hua, Z. Kira, Y. Shen, and H. Jin, "Continual diffusion: Continual customization of text-to-image diffusion with c-lora," *arXiv preprint arXiv:2304.06027*, 2023.

[25] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[26] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[27] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," *arXiv preprint arXiv:2307.01952*, 2023.