electronics

MDPI

Article

Black-Box Model Copyright Verification Based on Adaptive Boundary Fingerprints

Xiaorui Yu, Zhaozhong Guo * and Tianyi Zhang

College of Computer Science, Beijing Information Science and Technology University, Beijing 100192, China; yuxiaorui@bistu.edu.cn (X.Y.); 2025020733@bistu.edu.cn (T.Z.)
* Correspondence: guozhaozhong@bistu.edu.cn

Abstract

Deep-learning models are increasingly deployed in image recognition and intelligent decision-making, making model copyright verification important for protecting valuable digital assets. Existing structure- or parameter-based comparison methods are difficult to apply when only black-box access to a target model is available, and ordinary test samples often fail to reveal decision-boundary differences. This paper proposes a black-box copyright verification method based on adaptive boundary fingerprints. The method generates candidate boundary samples using the Fast Gradient Sign Method (FGSM), selects fingerprints under source correctness, margin, class balance, and diversity constraints, and measures source–target behavioral similarity by fusing label consistency, probability-distribution similarity, and boundary transition consistency. A decision threshold is then calibrated from the background distribution of independent models. Experiments on CIFAR-100 and Tiny-ImageNet show that independent models remain below calibrated thresholds, while most derived models obtain higher scores. On CIFAR-100, the method achieves a 91.7% detection rate and a 0.0% false-positive rate, with Area Under the Receiver Operating Characteristic Curve (AUROC) and Area Under the Precision-Recall Curve (AUPR) of 0.9750 and 0.9643, respectively. These results indicate that adaptive boundary fingerprints provide effective evidence for black-box model copyright verification.

Keywords: model copyright protection; model fingerprint; black-box verification; decision boundary

1. Introduction

In recent years, deep-learning models have been widely applied in areas such as image recognition, speech processing, natural language understanding, recommendation systems, and intelligent risk control. As model scale, training data volume, and computational demands continue to grow, the development of high-performance deep-learning models typically requires substantial investment in data collection, annotation, algorithm design, model training, and engineering tuning. Therefore, a trained deep-learning model is not only a core component of an algorithmic system but also a digital asset with clear commercial value and intellectual property attributes.

However, compared with traditional software systems, copyright protection for deep-learning models faces more complex challenges. First, model weight files can be directly copied and redeployed, allowing attackers to reuse the model owner's training outcomes without authorization, thereby directly infringing on model copyright and commercial interests. Second, even if attackers cannot directly access model parameters, they can

Check for updates

Academic Editor: Andreas Mauthe

Received: 7 June 2026
Revised: 4 July 2026
Accepted: 6 July 2026
Published: 13 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

construct functionally similar surrogate models through public API queries, training data sampling, or knowledge distillation, allowing model behavior to be indirectly copied and increasing the stealth of copyright infringement. Third, attackers can further fine-tune, prune, quantize, perform architecture transfer, or re-distill the stolen model to reduce structural and parametric similarity to the original, thereby evading traditional copyright detection methods and further complicating model copyright verification and infringement evidence collection. Such model theft and derivation not only harm the legitimate rights of model developers but also disrupt the normal order of the AI model market.

1.1. Research Problem

This paper focuses on the following problem: In a black-box scenario where only the target model's output can be queried and its network structure and parameters are inaccessible, can the model owner generate a set of boundary fingerprint samples from the source model and determine whether the target model is suspected to be derived from the source model based on the behavioral similarity between the source and target models on these samples?

This problem further involves three sub-questions:

First, how to construct boundary fingerprint samples that are more discriminative than ordinary test samples. Simply using adversarial examples as final fingerprints does not necessarily guarantee that samples are truly near the decision boundary, because adversarial perturbations may push samples across the boundary rather than keep them close to it. Thus, more explicit boundary selection criteria are needed.

Second, how to measure the boundary behavior similarity between the source model and the target model. Single hard-label prediction consistency is coarse and may miss confidence-level and transition-level differences, so multi-faceted detection scores are required.

Third, how to statistically calibrate the detection score based on the background distribution of independent models. Since unrelated models may also show certain behavioral similarity, a calibrated threshold is needed for more stable and interpretable decisions.

1.2. Contributions

The main contributions of this paper are as follows:

(1) We propose an adaptive boundary fingerprint construction method for black-box copyright verification. Since ordinary test samples fail to expose differences in model decision boundaries, we use the source model as a benchmark to generate candidate perturbed samples and apply constraints, including source-model correctness, classification margin, class balance, and sample diversity, to select the boundary fingerprint set, thereby enhancing the representativeness of the query samples for the local decision behavior of the source model.

(2) We design a multi-level boundary behavioral similarity metric and a statistical decision mechanism. Under black-box query conditions, we characterize the behavioral association between the source model and the target model from three levels: label consistency, probability-distribution similarity, and boundary transition consistency, and perform bootstrap quantile threshold calibration based on the sample-level background responses of independent models on the fingerprint set, thereby reducing false positives caused by natural behavioral similarity among independent models.

(3) We conduct systematic experimental evaluation under multiple derivation scenarios. On CIFAR-100 and Tiny-ImageNet, we construct target models including direct copy, fine-tuning, pruning, quantization, same-architecture boundary-enhanced distillation, and cross-architecture boundary-enhanced distillation, and compare with multiple

black-box fingerprint baselines. Experimental results show that our method achieves stable performance in fixed-threshold false-positive control and overall ranking ability, while also revealing that cross-architecture boundary-enhanced distillation can weaken local boundary-behavior inheritance and remains a major limitation of the current method.

1.3. Paper Structure

The remainder of this paper is organized as follows. Section 2 reviews related work on deep model copyright protection, focusing on model watermarking and model fingerprinting. Section 3 provides the necessary preliminary knowledge, including classification model outputs, black-box queries, decision boundaries, gradient perturbations, model fingerprints, and model copyright detection. Section 4 describes the threat model considered in this paper, clarifying the capabilities of the verifier, the attacker, and the ways of model derivation. Section 5 presents the proposed black-box model copyright verification method based on adaptive boundary fingerprints, including boundary fingerprint generation, behavioral similarity measurement, and threshold decision. Section 6 describes the experimental setup, and Section 7 presents and analyzes the experimental results. Section 8 concludes the paper and discusses limitations and future directions.

2. Related Work

Deep model copyright protection methods can be mainly divided into model watermarking and model fingerprinting [1–3]. Model watermarking methods typically embed specific ownership identifiers during model training, fine-tuning, or inference, and prove model ownership through white-box parameter inspection or black-box queries. Model fingerprinting methods do not modify the source model but instead construct query samples and compare the output behaviors of the source and target models to determine whether the two models exhibit unusually high similarity. This work belongs to the model fingerprinting approach, focusing on decision-boundary-based model derivation detection under black-box settings.

2.1. Model Watermarking Methods

Model watermarking is an early technical approach for deep model copyright protection. Uchida et al. [4] proposed embedding watermarks into deep neural network parameters by adding a regularization term to the training loss, so that model weights carry a specific binary signature. This method can verify model ownership in a white-box setting, but requires access to internal model parameters and typically needs the watermark to be pre-embedded during training, making it inapplicable to already-trained and unmodifiable source models.

To support black-box verification, Zhang et al. [5] extended traditional digital watermarking to deep neural networks by constructing specific trigger samples and embedding them into model behavior, allowing the verifier to query the trigger samples remotely to determine whether a model contains a watermark. Adi et al. [6] further used backdoor mechanisms to construct watermarks, making the model output a preset label on specific trigger inputs, thereby achieving black-box ownership verification. These methods are convenient for verification, but watermark embedding alters the training process and may be affected by operations such as fine-tuning, pruning, distillation, or trigger sample filtering.

Subsequently, Rouhani et al. [7] proposed DeepSigns, which embeds signatures into the probability distribution of model intermediate representations to enable both white-box and black-box model watermark verification. Chen et al. [8] proposed BlackMarks, using adversarial examples to generate watermark keys for black-box multi-bit watermark verifi-

cation. Szyller et al. [9] proposed Dynamic Adversarial Watermarking of Neural Networks (DAWN), dynamically modifying a small number of query responses at the model inference interface so that surrogate models extracted from the model inherit the watermark. These methods improve watermark capacity, black-box verifiability, or robustness against model extraction, but still rely on active embedding, model retraining, or inference interface control [10–13].

Overall, model watermarking methods have strong active ownership declaration capabilities, but their limitations are also evident: first, they typically require embedding during training or inference, making them unsuitable for already-trained models without pre-embedded watermarks; second, some methods may affect the original model behavior or introduce additional training costs; third, model modifications such as fine-tuning, pruning, quantization, and distillation can weaken the watermark [10,14,15]. Recent studies on large language model (LLM) watermarking also show that unauthorized distillation can weaken inherited watermark signals, highlighting distillation-based evasion risks [16]. Therefore, for already-trained models that can only be verified in a black-box manner, post hoc model fingerprinting methods are more applicable.

2.2. Model Fingerprinting Methods

Unlike model watermarking, model fingerprinting methods do not require modifying the source model; instead, they rely on the model's own input-output behavior as evidence of ownership. Such methods typically construct a set of query samples as a fingerprint set and compare the output consistency or behavioral similarity between the source model and the target model on that set. Since they do not depend on internal model structure or parameters, model fingerprints are more suitable for black-box copyright verification scenarios.

Cao et al. [17] proposed IPGuard, which extracts samples near the decision boundary of the source model as model fingerprints and judges model association based on label consistency on boundary samples, but it struggles to utilize fine-grained behaviors such as probability distributions and prediction transitions. Chen et al. [18] proposed DeepJudge, which constructs multiple types of test samples and compares model behavioral differences for copyright verification, but detection effectiveness depends on test sample generation and metric selection. Li et al. [19] proposed ModelDiff, which compares response change patterns on input pairs to characterize model reuse relationships, but it is more oriented towards general behavioral similarity comparison and does not specifically construct samples near the source model's decision boundary.

Recently, Xu et al. [20] proposed Adversarial Trajectory (ADV-TRA), using adversarial trajectory samples to characterize changes in model decision boundaries; Yang et al. [21] proposed AnaFP, analyzing fingerprint uniqueness and robustness from the perspective of fingerprint-to-boundary distance. These studies show that the decision boundary and its neighborhood behavior are important bases for black-box model copyright verification. DeepCore [22] further considers the distinction between homologous models and pirated models by constructing high-confidence core points for fingerprint verification. Bai et al. [23] proposed Steep-Decision-Boundary Fingerprinting (SDBF), which uses steep-decision-boundary fingerprints for hard-label tampering detection of deep neural network (DNN) models. In parallel, recent studies have further extended fingerprint-based ownership verification to LLMs, such as MergePrint, which studies merge-resistant black-box fingerprints for LLM ownership verification [24].

Existing model fingerprinting methods show that black-box input-output behavior can provide effective evidence for model copyright verification. Compared to ordinary test samples, samples near the boundary are more likely to expose differences in the

local discriminant functions of different models. However, directly using ordinary perturbed samples or a single-label consistency metric may be affected by sample distribution, class imbalance, perturbation strength, and background similarity of independent models. Therefore, how to construct more stable and discriminative boundary fingerprint sets and design finer-grained black-box behavioral similarity metrics remains a problem that model fingerprinting methods need to further address [25–29].

Motivated by these observations, our method adaptively selects boundary fingerprints by considering source-model correctness, boundary proximity, class balance, and feature-space diversity, and further combines label consistency, probability-distribution similarity, and boundary transition consistency with bootstrap threshold calibration. This design aims to improve the discriminability of boundary fingerprints while reducing the influence of sample-distribution bias, class imbalance, perturbation-strength sensitivity, redundant samples, single-metric bias, and background similarity among independent models.

3. Preliminaries

3.1. Classification Model and Output Representation

This paper considers multi-class classification tasks. Let $\mathcal{X}$ denote the input space and $\mathcal{Y} = \{1, 2, \ldots, C\}$ denote the class set, where C denotes the number of classes. Given an input sample $x \in \mathcal{X}$, a classification model f outputs a C-dimensional vector:

$$z = f(x) = (z_1, z_2, \ldots, z_C) \in \mathbb{R}^C, \tag{1}$$

where z_j represents the unnormalized output of the model for the j-th class, often called the logit. The predicted class of the model is:

$$\widehat{y}(x) = \arg\max_j z_j. \tag{2}$$

If a class probability distribution is needed, the softmax function converts logits into probabilities:

$$p_j(x) = \frac{\exp(z_j)}{\sum_{k=1}^{C} \exp(z_k)}. \tag{3}$$

Thus, the probability output of the model on sample x is:

$$p(x) = (p_1(x), p_2(x), \ldots, p_C(x)). \tag{4}$$

In supervised classification, cross-entropy loss is commonly used to measure the difference between the model output and the true label. For a sample (x, y), the cross-entropy loss is:

$$\mathcal{L}(f(x), y) = -\log p_y(x), \tag{5}$$

where $p_y(x)$ denotes the predicted probability assigned by the model to the true class y.

3.2. Black-Box Query and Output Formats

A black-box query means that the verifier cannot access the target model's network structure, parameters, or training process; they can only submit input samples to the model and receive output results returned via an interface. In model copyright detection scenarios, the black-box setting is closer to actual deployment environments, because the target model is typically deployed independently by a third party, and the model owner cannot directly inspect its internal information. Depending on the granularity of information returned by the interface, black-box outputs can be classified into two types.

(1) Label output. The model returns only the top-1 predicted class:

$$\widehat{y}(x) = \arg\max_{c} f(x)_c. \tag{6}$$

In this case, the verifier can only obtain the final predicted class and cannot observe the confidence scores for each class.

(2) Confidence output. The model returns a probability distribution or logits. In this case, the verifier can not only obtain the predicted class but also compare the confidence distributions of different models across classes. Compared with label output, confidence output contains finer-grained model behavior information and supports more detailed behavioral similarity measures.

3.3. Decision Boundary and Classification Margin

A classification model divides the input space into prediction regions corresponding to different classes; the separating surfaces between different class regions are called decision boundaries. If a sample is far from the decision boundary, the model typically makes a high-confidence prediction; if a sample is close to the decision boundary, a small perturbation may change the predicted class.

To characterize the relationship between a sample and the decision boundary, we can use the classification margin. Given a sample (x, y), let the logits of model f on sample x be $z = f(x)$. The classification margin is defined as:

$$m_f(x) = z_y - \max_{j \neq y} z_j. \tag{7}$$

When $m_f(x)$ is large, the model has a clear discriminative advantage for the true class, and the sample is typically in a high-confidence prediction region; when $m_f(x)$ is close to 0, the gap between the true class and competing classes is small, and the sample is more likely to be near the decision boundary.

3.4. Gradient Perturbation and Fast Gradient Sign Method

Gradient perturbation uses the gradient of the loss function with respect to the input sample to move the input in the direction that increases the model's loss. Given a model f, a sample (x, y), and a loss function $\mathcal{L}$, the input gradient is:

$$\nabla_x \mathcal{L}(f(x), y). \tag{8}$$

The Fast Gradient Sign Method (FGSM) is a one-step gradient perturbation method of the form:

$$x^{'} = \text{clip}(x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f(x), y))), \tag{9}$$

where ϵ is the perturbation strength, $\text{sign}(\cdot)$ denotes the sign function, and $\text{clip}(\cdot)$ restricts the perturbed sample to the valid input range. FGSM requires only one backpropagation step, making it computationally inexpensive and suitable for generating candidate perturbed samples.

3.5. Model Fingerprint and Behavioral Similarity

A model fingerprint refers to a set of query samples and their corresponding responses used to characterize the input-output behavior of a model. Let $\mathcal{Q}$ denote the set of query samples:

$$\mathcal{Q} = \{q_i\}_{i=1}^{K}. \tag{10}$$

For a model f, the observable responses on this set can be expressed as:

$$R_f(\mathcal{Q}) = \left\{ r_f(q_i) \right\}_{i=1}^{K}. \tag{11}$$

Here, $r_f(q_i)$ can be a predicted label, a probability distribution, or logits. Given two models f_a and f_b, their behavioral similarity on query set $\mathcal{Q}$ can be abstractly expressed as:

$$S(f_a, f_b; \mathcal{Q}) = Sim\Big(R_{f_a}(\mathcal{Q}), R_{f_b}(\mathcal{Q})\Big), \tag{12}$$

where $Sim(\cdot, \cdot)$ denotes a general response-similarity function. In the proposed method, $Sim(\cdot, \cdot)$ is not instantiated as a single similarity metric; instead, it is later defined as a composite behavioral similarity measure that integrates label consistency, probability-distribution similarity, and boundary transition consistency. The core idea of model fingerprinting methods is to compare model behaviors using a specifically designed query set, thereby providing evidence for model relationship analysis.

3.6. Model Copyright Detection

Model copyright detection aims to determine whether a target model is likely to be copied, modified, or indirectly trained from a source model held by the model owner. Let f_s be the source model and f_t be the target model. If f_t inherits part of the parameters, knowledge, or functional behavior of f_s through direct copying, fine-tuning, knowledge distillation, pruning, quantization, or other means, it can be regarded as a suspected derived model of the source model.

In a white-box scenario, the verifier can compare model structures, parameters, or intermediate representations; in a black-box scenario, the verifier cannot access internal information of the target model and can only obtain model outputs through a query interface. Therefore, black-box model copyright detection typically relies on a set of query samples and their output responses to measure the behavioral similarity between the source model and the target model.

4. Threat Model

This paper considers scenarios where derived models are constructed for model copyright verification. The attacker's goal is to reuse the knowledge or functional behavior of the source model without authorization and construct an independently deployable target model. Depending on the attacker's level of access to the source model and ability to modify it, the ways of constructing a target model can be divided into three categories: direct parameter inheritance, behavior mimicry learning, and model post-processing transformations. Specifically, this paper considers the following typical model derivation methods:

(1) Direct copy: The attacker obtains the source-model weights and deploys them directly, resulting in a model with identical structure and parameters to the source-model.
(2) Fine-tuning: The attacker initializes from the source model and continues training on part or all of the training data, altering model parameters while maintaining high task performance.
(3) Knowledge distillation: The attacker uses the source model's outputs as teacher signals to train a student model, which may have the same or a different architecture.
(4) Pruning: The attacker performs structured or unstructured pruning on the source model, possibly followed by restorative fine-tuning.
(5) Quantization: The attacker reduces the precision of the source model's parameters or activations to low-bit representations, changing the parameter distribution to facilitate deployment.

We assume that the attacker can perform the above copying, training, and post-processing operations on the source model and can modify the structure and parameters of the target model. However, the final sample-level fingerprint set is selected on the owner side from owner-accessible held-out validation data according to source-model

boundary behavior, margin, class balance, and feature-space diversity constraints, and is not disclosed to the attacker. Therefore, even if the attacker knows the general use of FGSM-based boundary candidates and adaptive filtering, the concrete validation samples, selected fingerprints, and calibrated threshold remain unavailable for direct optimization in the current threat model [30].

5. Method Description

We propose a black-box model copyright verification method based on adaptive boundary fingerprints. The method generates boundary fingerprint samples using the source model as a benchmark, and determines whether the target model is suspected to be derived from the source model by comparing the behavioral similarity of their outputs on the fingerprint samples.

The proposed method comprises five consecutive stages: candidate boundary sample generation, boundary fingerprint selection, model behavior query, similarity score computation, and threshold decision. The overall process is illustrated in Figure 1.

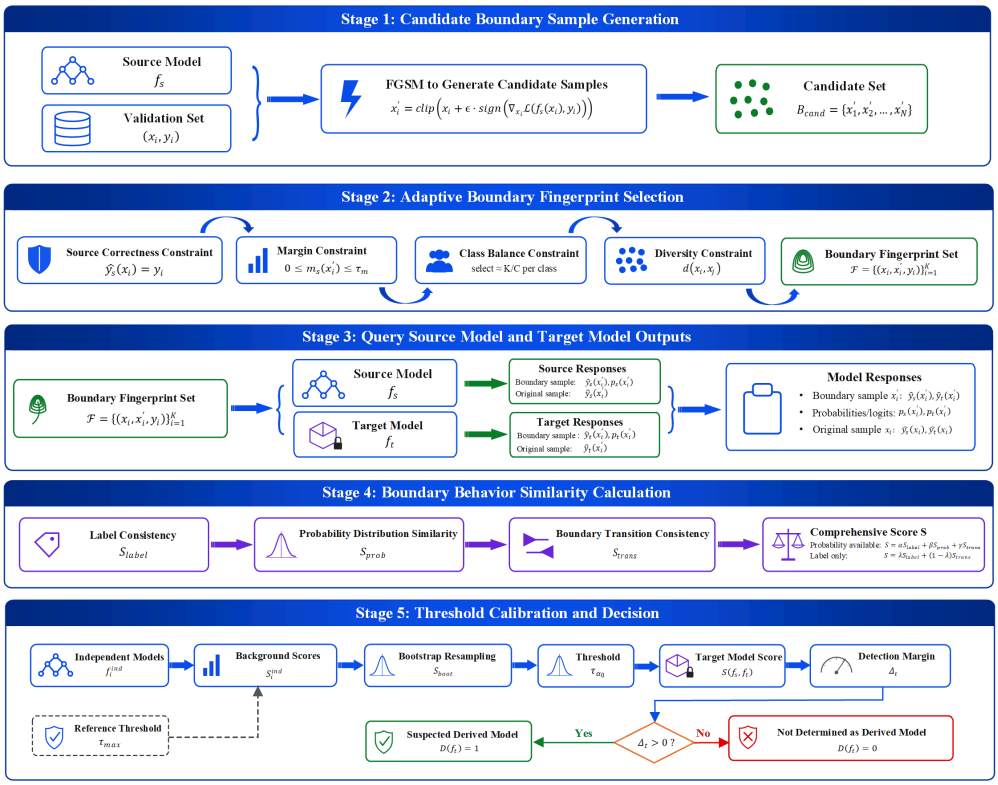


Figure 1. Flowchart of black-box model copyright verification based on adaptive boundary fingerprints.

5.1. Method Workflow

The overall data flow of the proposed method is as follows:

$$\mathcal{D}_{\text{val}} \longrightarrow \mathcal{B}_{\text{cand}} \longrightarrow \mathcal{F} \longrightarrow \mathcal{R}_s(\mathcal{F}), \mathcal{R}_t(\mathcal{F}) \longrightarrow S(f_s, f_t) \longrightarrow D(f_t), \tag{13}$$

where $\mathcal{D}_{\text{val}}$ denotes an owner-accessible held-out validation dataset distinct from the source training set, $\mathcal{B}_{\text{cand}}$ denotes the set of candidate boundary samples generated by the source model, $\mathcal{F}$ denotes the adaptively selected boundary fingerprint set, $\mathcal{R}_s(\mathcal{F})$ and $\mathcal{R}_t(\mathcal{F})$ denote the output responses of the source model and the target model on the fingerprint set, $S(f_s, f_t)$ denotes the combined detection score, and $D(f_t)$ denotes the final decision.

Specifically, in the experimental setup, $\mathcal{D}_{\text{val}}$ is constructed as a held-out split from the public benchmark datasets and is used only for query sample generation and fingerprint

selection, rather than for source-model training or target-model construction. In practical deployment, it corresponds to validation or calibration data accessible to the legitimate model owner, which may be privately retained by the owner or legally accessible auxiliary data. The verifier first uses the source model f_s as a benchmark to apply gradient perturbations to samples from $\mathcal{D}_{\text{val}}$, generating a candidate boundary sample set $\mathcal{B}_{\text{cand}}$. Then, the verifier selects from the candidate samples a boundary fingerprint set $\mathcal{F}$ according to four constraints: source-model prediction correctness, closeness to the source model's decision boundary, relatively balanced class distribution, and low feature-space redundancy measured by the distance between source-model feature representations. In this selection process, samples misclassified by the source model are first excluded. Among the remaining correctly classified samples, the margin-based boundary constraint gives priority to samples with smaller positive margins, since they are closer to the source model's decision boundary while still preserving the source model's correct prediction.

During verification, the verifier queries both the source model f_s and the target model f_t using the same fingerprint set. In the full verification setting, we assume access to confidence or probability outputs. The label consistency and boundary transition consistency metrics use the top-1 predicted labels derived from the probability vectors, while the probability-distribution similarity metric directly uses the full probability vectors. If logits are returned, they can be converted into probabilities by softmax; if only hard labels are available, the method falls back to the label-only degraded score described later. The verifier then computes a combined detection score based on label consistency, probability-distribution similarity, and boundary transition consistency, which together characterize the source model's boundary-neighborhood decision behavior. Finally, the verifier constructs a background distribution of independent models and performs bootstrap quantile threshold calibration to make a decision on the target model.

Through the above process, we transform black-box model copyright verification into a model behavioral similarity comparison problem on a boundary fingerprint set. Compared with ordinary test samples, boundary fingerprint samples are closer to the source model's decision boundary, thus better reflecting whether the source model's local discriminative behavior is inherited by the target model. Compared with using a fixed empirical threshold, threshold calibration based on the background distribution of independent models reduces the risk of false positives due to natural similarity among independent models.

5.2. Method Steps

Step 1: Generate candidate boundary samples.

Given a source model f_s and a validation sample (x_i, y_i), we generate candidate boundary samples using the FGSM formulation introduced in Equation (9). Specifically, the input gradient is computed according to Equation (8), with the general model f instantiated as the source model f_s. The resulting perturbed sample is denoted as

$$x_i^{'} = \text{FGSM}_{\epsilon}(f_s, x_i, y_i), \tag{14}$$

where $\text{FGSM}_{\epsilon}(\cdot)$ follows the perturbation rule defined in Equation (9). This step uses FGSM to efficiently construct a source-guided candidate boundary sample set $\mathcal{B}_{\text{cand}}$; the final fingerprints are selected later through the adaptive boundary selection procedure.

Step 2: Select adaptive boundary fingerprints.

Taking the candidate boundary sample set $\mathcal{B}_{\text{cand}}$ as input, we sequentially apply the source-model correctness constraint, classification margin constraint, class balance constraint, and sample diversity constraint. This sequence is designed to make the selected fingerprints both boundary-relevant and representative. Specifically, source correctness

removes samples for which the source model is unreliable, and the margin constraint focuses the candidates on the decision-boundary region. Class balance then prevents the fingerprint set from being dominated by a few classes, while diversity screening reduces feature-space redundancy. These constraints also involve trade-offs: stricter margin filtering may reduce class coverage, whereas stronger diversity screening may include samples with slightly weaker boundary proximity.

First, we apply the correctness constraint, keeping only samples for which the source model predicts correctly on the original sample. The predicted label $\widehat{y}_s(x_i)$ is obtained according to the label prediction rule in Equation (2). A sample is retained if

$$\widehat{y}_s(x_i) = y_i, \tag{15}$$

and is discarded otherwise. This constraint excludes anomalous samples that the source model already misclassifies.

Second, we apply the classification margin constraint to the perturbed sample $x_i^{'}$. The source-model margin $m_s(x_i^{'})$ is computed according to the classification margin definition in Equation (7), with the general model f instantiated as the source model f_s and the reference label set to y_i. We preferentially retain samples satisfying:

$$0 \leq m_s\left(x_i^{'}\right) \leq \tau_m, \tag{16}$$

where τ_m is the classification margin threshold. This condition selects samples that are close to the source model's decision boundary but have not crossed it. If the number of samples satisfying this condition is insufficient, we supplement the fingerprint set by selecting additional non-crossing candidates with $m_s(x_i^{'}) \geq 0$ in ascending order of classification margin.

Third, we apply the class balance constraint to avoid fingerprint samples being concentrated in a few classes. Let C be the number of classes and K be the target number of fingerprint samples. We use approximately $\frac{K}{C}$ samples per class as a soft quota rather than a strict requirement. If a class has fewer valid boundary candidates after the previous filtering steps, all available candidates from that class are retained, and the remaining positions are filled from the other candidates according to the margin and diversity criteria.

Finally, we apply the sample diversity constraint to reduce feature-space redundancy among samples of the same class. Let $\phi_s(\cdot)$ denote the feature extractor of the penultimate layer of the source model. The feature distance between two samples is defined as:

$$d\left(x_i, x_j\right) = \| \phi_s(x_i) - \phi_s\left(x_j\right) \|_2 . \tag{17}$$

A smaller feature distance indicates higher redundancy, while a larger feature distance indicates stronger sample diversity. Within each class, we preferentially select samples that have a small classification margin and are more dispersed in the feature space. After the above selection, we obtain the final boundary fingerprint set:

$$\mathcal{F} = \left\{ \left(x_i, x_i^{'}, y_i \right) \right\}_{i=1}^{K}. \tag{18}$$

Step 3: Query source and target model outputs.

Feed the boundary fingerprint set $\mathcal{F}$ into the source model f_s and the target model f_t to obtain their output responses on the boundary samples. For a boundary fingerprint sample $x_i^{'}$, the predicted labels of the source and target models, denoted as $\widehat{y}_s(x_i^{'})$ and $\widehat{y}_t(x_i^{'})$, are obtained according to the label prediction rule in Equation (2).

If the target model's interface returns a probability distribution or logits, we further obtain the probability outputs $p_s(x_i^{'})$ and $p_t(x_i^{'})$. When logits are returned, they are converted into probability distributions using the softmax transformation introduced in Equation (3). If boundary transition consistency is needed, we also query the predictions of the source and target models on the original samples x_i. This step outputs the response sets of the source and target models on the boundary fingerprint set.

Step 4: Compute boundary behavioral similarity.

We measure the behavioral similarity between the source model and the target model from three aspects: label consistency, probability-distribution similarity, and boundary transition consistency.

Before computing the following similarity scores, the final fingerprint set $\mathcal{F}$ is required to satisfy a minimum viable cardinality. Let N_{tar} denote the target fingerprint size. We define $N_{\min} = \lceil 0.1 N_{\text{tar}} \rceil$. In our experiments, $N_{\text{tar}} = 500$, and thus $N_{\min} = 50$. If the adaptive selection procedure produces fewer than N_{tar} but at least $N_{\min}$ valid fingerprint samples, all available valid samples are used for score computation. If $|\mathcal{F}| < N_{\min}$, the similarity scores are not computed, and the candidate generation and selection procedure is repeated with an enlarged candidate pool or adjusted perturbation strength.

Label consistency measures whether the predicted classes of the two models on boundary fingerprint samples are consistent. It is defined as:

$$S_{\text{label}} = \frac{1}{|\mathcal{F}|} \sum_{\left(x_i, x_i^{'}, y_i\right) \in \mathcal{F}} \mathbf{1}\left[\widehat{y}_s\left(x_i^{'}\right) = \widehat{y}_t\left(x_i^{'}\right)\right], \tag{19}$$

where $\mathbf{1}[\cdot]$ is the indicator function, taking 1 if the condition holds and 0 otherwise. This metric reflects the hard-label behavior consistency of the source and target models on boundary samples.

Probability-distribution similarity supplements the information missed by label consistency, capturing differences in confidence distributions and providing a finer-grained characterization of output behavior on boundary samples. When probability outputs are available, we use the Jensen-Shannon (JS) divergence to compute probability-distribution similarity, defined as:

$$S_{\text{prob}} = 1 - \frac{1}{|\mathcal{F}|} \sum_{\left(x_i, x_i^{'}, y_i\right) \in \mathcal{F}} JS\left(p_s\left(x_i^{'}\right), p_t\left(x_i^{'}\right)\right). \tag{20}$$

We use base-2 logarithms to compute the JS divergence so that its range is $[0, 1]$, ensuring S_{prob} is also in $[0, 1]$.

Boundary transition consistency measures whether the prediction changes of the two models from the original sample to the boundary sample are consistent. Define the prediction transition pairs of the source model and the target model as:

$$T_s(x_i, x_i^{'}) = \left(\widehat{y}_s(x_i), \widehat{y}_s(x_i^{'})\right), \tag{21}$$

$$T_t(x_i, x_i^{'}) = \left(\widehat{y}_t(x_i), \widehat{y}_t(x_i^{'})\right). \tag{22}$$

Then, boundary transition consistency is:

$$S_{\text{trans}} = \frac{1}{|\mathcal{F}|} \sum_{\left(x_i, x_i^{'}, y_i\right) \in \mathcal{F}} \mathbf{1}\left[T_s\left(x_i, x_i^{'}\right) = T_t\left(x_i, x_i^{'}\right)\right]. \tag{23}$$

When probability outputs are available, the combined detection score is defined as:

$$S = \alpha S_{\text{label}} + \beta S_{\text{prob}} + \gamma S_{\text{trans}}, \tag{24}$$

$$\alpha + \beta + \gamma = 1, \tag{25}$$

where α, β, and γ are the weights for label consistency, probability distribution similarity, and boundary transition consistency, respectively.

In scenarios where only label output is available (probability distribution cannot be obtained), the combined detection score reduces to:

$$S = \lambda S_{\text{label}} + (1 - \lambda) S_{\text{trans}}, \tag{26}$$

where λ is the weight between label consistency and boundary transition consistency.

Step 5: Threshold calibration and decision.

To determine whether the target model's score is abnormally high, we construct background scores for non-derived models using multiple independent models. For each independent model f_i^{ind}, we compute its combined detection score with the source model using the same boundary fingerprint set $\mathcal{F}$:

$$S_i^{\text{ind}} = S(f_s, f_i^{\text{ind}}), \quad i = 1, 2, \ldots, M. \tag{27}$$

Thus, we obtain the background score set of independent models:

$$S_{\text{ind}} = \left\{ S_1^{\text{ind}}, S_2^{\text{ind}}, \ldots, S_M^{\text{ind}} \right\}. \tag{28}$$

To reduce the sensitivity of threshold estimation to a specific fingerprint sample set, we employ bootstrap resampling to estimate the upper-tail quantile of the independent-model background scores. We perform B bootstrap resampling iterations on the fingerprint sample dimension, compute the average score for each independent model on the resampled fingerprint set, and aggregate all resampled background average scores into S_{boot}. Given a target false-positive rate α_0, the formal decision threshold is defined as:

$$\tau_{\alpha_0} = Q_{1-\alpha_0}(S_{boot}). \tag{29}$$

For a target model f_t, we define the detection margin as:

$$\Delta_t = S(f_s, f_t) - \tau_{\alpha_0}. \tag{30}$$

If the detection margin satisfies:

$$\Delta_t > 0, \tag{31}$$

then the target model is determined to be a suspected derived model of the source model; otherwise, the current behavioral evidence is considered insufficient. The decision function can be expressed as:

$$D(f_t) = \begin{cases} 1, & \Delta_t > 0, \\ 0, & \Delta_t \leq 0. \end{cases} \tag{32}$$

Additionally, we compute the maximum-background score threshold:

$$\tau_{\max} = \max_{1 \leq i \leq M} S_i^{\text{ind}}. \tag{33}$$

This threshold, representing the maximum-background score among independent models, is used as an empirical maximum-background reference threshold to examine whether the target model's score not only exceeds the bootstrap statistical threshold but also surpasses all observed independent-model background scores. Our formal decision uses τ_{α_0} and the detection margin Δ_t, while $\tau_{\max}$ is used only for reference analysis.

5.3. Complexity Analysis

The proposed method is a black-box model copyright verification method, and the online verification stage does not involve model training. The independent models are used only in the offline stage to construct the non-derived model background distribution and calibrate the decision threshold. The time and space complexity of each stage are summarized in Table 1, where M denotes the number of independent models, K denotes the number of boundary fingerprint samples, N_v denotes the number of candidate validation samples, T_{div} denotes the cost of feature-distance computation in diversity-based filtering, C denotes the number of classes, B denotes the number of bootstrap resampling iterations, T_{ind} denotes the time cost of training a single independent model, P denotes the number of parameters of a single model, and d_x denotes the input sample dimension.

Table 1. Complexity analysis of different stages of the proposed method.

Stage	Main Operation	Time Complexity	Space Complexity
Offline background model construction	Train M independent models	$O(MT_{\text{ind}})$	Peak $O(P)$; $O(MP)$ if all models are stored
Offline fingerprint generation	Generate candidate boundary samples using FGSM	$O(N_v)$ gradient computations	$O(N_v d_x)$
Offline fingerprint selection	Correctness filtering, margin filtering, class balancing, and diversity filtering	$O(N_v \log N_v + T_{\text{div}})$	$O(Kd_x)$
Offline background score computation	Query M independent models using the fingerprint set	$O(MK)$ model queries; probability-distribution comparison costs $O(MKC)$	$O(MKC)$ for full probability outputs; $O(MK)$ for sample-level scores
Offline threshold calibration	Bootstrap resampling	$O(BMK)$	$O(MK + BM)$
Online model verification	Query the target model and compute the detection score	$O(K)$ or $O(2K)$ model queries; probability-distribution comparison costs $O(KC)$	$O(KC)$ or $O(K)$

In terms of time complexity, the main additional cost of the proposed method comes from the offline stage, especially the training of independent models. However, this process is used only to construct the background distribution and calibrate the decision threshold, and does not need to be repeated for each target model. Given a task, a source model, and a background data setting, the independent-model background distribution, the boundary fingerprint set, and the decision threshold can be reused for verifying multiple target models. The online verification cost mainly comes from black-box queries. If only boundary fingerprint samples are queried, K target-model queries are required. If boundary transition consistency is also computed, both the original samples and boundary samples are queried, resulting in approximately $2K$ target-model queries. Therefore, the online verification cost is approximately linear in the number of fingerprint samples and does not involve model training.

In terms of space complexity, the proposed method does not require all independent models to be stored permanently. In practice, independent models can be trained sequentially, and each model checkpoint can be deleted after computing its output responses or background scores on the fingerprint set. Therefore, the peak model-storage cost during offline calibration is approximately $O(P)$, rather than $O(MP)$. After threshold calibration is completed, the main long-term storage requirements are the boundary fingerprint set,

the source-model responses, the independent-model background scores, and the final decision threshold.

Overall, in the medium-scale image classification copyright verification scenario considered in this paper, the proposed method is practically feasible. For larger-scale models or commercial deployment scenarios with higher query costs, the number of independent models and the number of fingerprint samples can be further adjusted according to the available computational resources, so as to balance threshold calibration stability and verification cost.

5.4. Method Summary

Algorithm 1 summarizes the complete procedure of the proposed method. First, the algorithm uses the source model and validation data to generate candidate boundary samples, then adaptively selects the boundary fingerprint set. Second, it queries the output behaviors of the source model, target model, and independent models using this set and computes the combined detection score. Finally, it calculates the detection margin based on the bootstrap quantile threshold τ_{α_0} and makes a decision; the maximum-background score $\tau_{\max}$ is used only as an empirical reference.

Through the above algorithm steps, we transform black-box model copyright verification into a behavioral similarity comparison problem on the same boundary fingerprint set.

Algorithm 1 Black-box model copyright verification based on adaptive boundary fingerprints.

Input: Source model f_s; target model f_t; validation dataset $\mathcal{D}_{\text{val}}$; independent models $\{f_i^{\text{ind}}\}_{i=1}^{M}$; perturbation strength ϵ; margin threshold τ_m; target fingerprint size K; score weights α, β, γ; target false-positive rate α_0; number of bootstrap resamples B.

Output: Boundary fingerprint set $\mathcal{F}$; detection score $S(f_s, f_t)$; statistical threshold τ_{α_0}; maximum-background threshold $\tau_{\max}$; detection margin Δ_t; decision $D(f_t)$.

$\mathcal{B}_{\text{cand}} \leftarrow \varnothing$.
for each $(x_i, y_i) \in \mathcal{D}_{\text{val}}$ **do**
  if $\widehat{y}_s(x_i) \neq y_i$ **then**
    continue
  end if
  $x_i' \leftarrow \text{FGSM}_\epsilon(f_s, x_i, y_i)$.
  $m_s(x_i', y_i) \leftarrow z_{s,y_i}(x_i') - \max_{j \neq y_i} z_{s,j}(x_i')$.
  if $0 \leq m_s(x_i', y_i) \leq \tau_m$ **then**
    $\mathcal{B}_{\text{cand}} \leftarrow \mathcal{B}_{\text{cand}} \cup \{(x_i, x_i', y_i)\}$.
  end if
end for
if $|\mathcal{B}_{\text{cand}}| < K$ **then**
  Supplement candidate samples in ascending order of classification margin.
end if
$\mathcal{F} \leftarrow \text{Select}(\mathcal{B}_{\text{cand}}, K)$, where $\text{Select}(\cdot)$ applies class balance and feature diversity screening.
Compute the sample-level score vector $\mathbf{a}_t = \{a_{t,k}\}_{k=1}^{K}$ between f_s and f_t on $\mathcal{F}$, where $a_{t,k} = \alpha a_{t,k}^{\text{label}} + \beta a_{t,k}^{\text{prob}} + \gamma a_{t,k}^{\text{trans}}$.
$S(f_s, f_t) \leftarrow \frac{1}{K} \sum_{k=1}^{K} a_{t,k}$.
$A^{\text{ind}} \leftarrow \varnothing$, $S_0 \leftarrow \varnothing$.
for $i = 1$ to M **do**

Algorithm 1 *Cont.*

Compute the sample-level score vector $\mathbf{a}_i^{\text{ind}} = \{a_{i,k}^{\text{ind}}\}_{k=1}^{K}$ between f_s and f_i^{ind} on $\mathcal{F}$.
$S_i^{\text{ind}} \leftarrow \frac{1}{K}\sum_{k=1}^{K} a_{i,k}^{\text{ind}}$.
$A^{\text{ind}} \leftarrow A^{\text{ind}} \cup \{\mathbf{a}_i^{\text{ind}}\}$, $S_0 \leftarrow S_0 \cup \{S_i^{\text{ind}}\}$.
end for
$S_{boot} \leftarrow \text{Bootstrap}_k(A^{\text{ind}}, B)$, where Bootstrap_k resamples fingerprint indices k with replacement and averages the corresponding sample-level scores for each independent model.
$\tau_{\alpha_0} \leftarrow Q_{1-\alpha_0}(S_{boot})$.
$\tau_{\max} \leftarrow \max_{1\le i\le M} S_i^{\text{ind}}$.
$\Delta_t \leftarrow S(f_s, f_t) - \tau_{\alpha_0}$.
if $\Delta_t > 0$ **then**
$D(f_t) \leftarrow 1$.
else
$D(f_t) \leftarrow 0$.
end if
return $\mathcal{F}$, $S(f_s, f_t)$, τ_{α_0}, $\tau_{\max}$, Δ_t, $D(f_t)$.

6. Experimental Design

Our experiments focus on the effectiveness, stability, and applicability of black-box model copyright verification. The core objective is to verify whether, under conditions where the internal structure and parameters of the target model are inaccessible, the adaptive boundary fingerprints constructed from the source model can effectively distinguish source-model-derived models from independently trained models.

We first construct a source model, multiple independent models, and various types of target-derived models. Then, we compute the combined detection score between the source model and target models using the adaptive boundary fingerprints to test the detection capability of our method under derivation scenarios, including direct copy, fine-tuning, pruning, quantization, and boundary-enhanced distillation. Subsequently, we conduct ablation experiments, robustness evaluations, and parameter sensitivity analyses from the perspectives of query sample types, baseline method comparison, input-level preprocessing defenses, restricted hard-label output settings, fingerprint sample size, perturbation strength, classification margin threshold, and other factors, to further evaluate the impact of different design and deployment factors on detection performance. The overall experimental design covers dataset and model settings, construction of target models, construction of fingerprint samples, evaluation metrics, and implementation details.

6.1. Datasets

We use two image classification datasets: CIFAR-100 and Tiny-ImageNet. CIFAR-100 contains 100 fine-grained classes, with images of size 32×32, 50,000 training images, and 10,000 test images. Tiny-ImageNet contains 200 classes, with images of size 64×64, 100,000 training images, and 10,000 validation images. Both datasets are multi-class image classification tasks with relatively complex class boundaries, suitable for evaluating the ability of boundary fingerprint samples to characterize the local decision behavior of models.

6.2. Experimental Environment and Parameter Settings

All experiments are conducted in a unified software and hardware environment. Model training, derived model construction, fingerprint sample generation, and black-box queries are performed on the same platform to minimize the impact of environmental differences on experimental results. The specific experimental environment configuration is shown in Table 2.

Table 2. Experimental environment configuration.

Configuration	Setting
Operating System	Ubuntu 22.04.5 LTS
CPU	Intel Xeon Platinum 8470Q
GPU	NVIDIA GeForce RTX 5090, 32 GB GPU memory
Python Version	3.12.3
Deep-Learning Framework	PyTorch 2.8.0
CUDA Version	12.8

Unless otherwise specified, the main experiments use the default hyperparameter settings listed in Table 3.

Table 3. Default experimental hyperparameter settings.

Parameter	Default Setting	Description
Fingerprint size K	500	Number of adaptive boundary fingerprints used for black-box verification
Perturbation strength ϵ	8/255	Perturbation magnitude used to generate candidate boundary samples
Margin threshold τ_m	1.0	Threshold for selecting samples close to the source-model decision boundary
Combined score weights α, β, γ	1/3, 1/3, 1/3	Default equal-weight fusion of label consistency, probability-distribution similarity, and boundary transition consistency
Number of independent models M	30	Number of independently trained models used to estimate the non-derived background distribution
Target false-positive rate α_0	0.05	Target false-positive level used for bootstrap quantile threshold calibration
Bootstrap iterations B	1000	Number of bootstrap resamples used for threshold calibration based on independent-model background responses

Through the above experimental setup, we systematically evaluate the effectiveness of the adaptive boundary fingerprint method for black-box model copyright verification in terms of dataset generalization, derivation coverage, query sample construction, strong baseline comparison, threshold calibration, and parameter stability.

6.3. Models and Verification Object Construction

We divide the experimental models into three categories: source model, independent models, and target-derived models. The source model is the model to be protected, owned by the model owner, used to generate boundary fingerprints and provide baseline outputs. Independent models are models independently trained on the same task as the source model but without inheriting its parameters or output behavior; they are used to estimate the background distribution of non-derived models. Target-derived models are obtained from the source model through copying, fine-tuning, pruning, quantization, or distillation, and are used to simulate various forms of model theft, compression deployment, and model extraction scenarios.

Unless otherwise specified, we use ResNet18 as the base model architecture on both datasets. Although ResNet18 is used as the main backbone for controlled evaluation, the proposed verification framework is not specific to ResNet18. Boundary fingerprint generation only requires input gradients from the source model, while target-model verification

relies on black-box output responses. Therefore, the framework can, in principle, be applied to other differentiable classification architectures. However, architectural changes may affect how much local boundary behavior is inherited by the derived model, especially in cross-architecture distillation settings. For each dataset, we train one source model and 30 independent models. All 30 independent models are initialized and trained with different random seeds, without loading source-model parameters or using source-model outputs as supervision, ensuring that they serve as valid estimates of the background distribution of non-derived models. Thus, the current background set mainly captures the same-architecture variation caused by independent initialization and training stochasticity, while broader architectural and training-strategy diversity is left for future work. The architectures and construction methods for each model type are shown in Table 4. For compact presentation in the model names used in the following tables, fine-tuning and knowledge distillation are abbreviated as FT and KD, respectively.

Table 4. Architectures and construction methods for non-distillation model types.

Model Type	Architecture	Number and Setting	Construction Method
Source	ResNet18	1	Train on the corresponding training set
Independent	ResNet18	30	Train independently with different random seeds
Copy	ResNet18	1	Directly load the source-model weights
FT-All-10/50/100%	ResNet18	CIFAR-100: 3; Tiny-ImageNet: 1 (10%)	Fine-tune using auxiliary data
Prune-40/60/80%	ResNet18	CIFAR-100: 3; Tiny-ImageNet: 1 (40%)	Apply unstructured pruning
Quant-8bit/4bit	ResNet18	CIFAR-100: 2; Tiny-ImageNet: 1 (8-bit)	Apply weight quantization

Fine-tuned models use full-model fine-tuning to examine whether the source model's boundary behavior remains detectable after the attacker performs overall parameter updates. Pruned models use different pruning ratios followed by restorative fine-tuning to analyze the impact of structural sparsification on boundary fingerprints. Quantized models are constructed by applying post-training weight-only fake quantization to the trained source model. Specifically, we use symmetric uniform per-tensor quantization for the weights and biases of convolutional and fully connected layers at 8-bit and 4-bit precision, without activation quantization or quantization-aware training.

For distillation models, we focus on boundary-enhanced distillation and cross-architecture boundary-enhanced distillation. Boundary-enhanced distillation further adds boundary-perturbed samples generated by the source model on top of natural sample distillation to analyze whether the student model inherits the source model's local boundary behavior. The distillation data scale denotes the ratio between the number of auxiliary distillation samples and the size of the source training set. Specifically, 0.6 means that the auxiliary distillation data contain 60% as many samples as the source training set, while 1.0 means that the auxiliary distillation data have the same size as the source training set. Thus, 0.6 simulates medium-scale in-distribution auxiliary data, and 1.0 constructs a high-data boundary distillation setting. Cross-architecture boundary distillation uses MobileNetV2 as

the student model to analyze the impact of student model structural changes on detection difficulty. The construction of distillation models is shown in Table 5.

Table 5. Architectures and construction methods for distillation-derived models.

Distillation Model	Teacher Model	Student Model	Distillation Data Scale	Distillation Sample Type
KD-ResNet18-Boundary-0.6	Source ResNet18	ResNet18	0.6× source training set size	Natural samples + boundary-perturbed samples
KD-ResNet18-Boundary-1.0	Source ResNet18	ResNet18	1.0× source training set size	Natural samples + boundary-perturbed samples
KD-MobileNet-Boundary-1.0	Source ResNet18	MobileNetV2	1.0× source training set size	Natural samples + boundary-perturbed samples

The distillation data scale indicates the size of the in-distribution auxiliary data available to the attacker relative to the source training set. In the public dataset experiments, we use different proportions of the source dataset as a proxy for in-distribution auxiliary data to control variables and analyze the impact of distillation intensity on detection scores.

6.4. Query Samples and Baseline Settings

To analyze the impact of query sample construction on black-box copyright verification effectiveness, we set five types of query samples: Clean, Random, FGSM, Projected Gradient Descent (PGD), and Adaptive FGSM. All query samples are constructed based on the same source model and the same validation data, differing in whether perturbations are added, whether the perturbation direction is guided by the source-model gradient, and whether boundary and diversity screening is applied. The query sample types and construction methods are shown in Table 6.

Table 6. Query sample types and construction methods.

Query Sample Type	Construction Method	Utilizes Source-Model Gradient?	Adaptive Screening?
Clean	Directly select ordinary test/validation samples	No	No
Random	Add random perturbation to ordinary samples	No	No
FGSM	One-step perturbed samples along source model loss increase direction	Yes	No
PGD	Multi-step projected gradient perturbed samples	Yes	No
Adaptive FGSM	FGSM samples screened by correctness, margin, class balance, and diversity	Yes	Yes

Adaptive boundary fingerprint samples are obtained by screening from the FGSM candidate samples. If the strict screening conditions yield insufficient samples, we expand the candidate pool or supplement samples by sorting by classification margin; in extreme cases, we use all actually screened samples and report the actual fingerprint size in the experimental results.

To ensure a controlled and fair comparison, we design the baseline comparison under a unified experimental protocol. The source model, target model set, independent-model background distribution, and threshold calibration strategy are kept consistent across all compared methods. The query budget is also controlled to be the same or of the same order of magnitude, so that the comparison mainly reflects differences in query construction

and similarity measurement rather than differences in model selection, query scale, or threshold tuning.

Table 7 provides an overview of the comparison methods considered in this experiment. Each baseline is adapted to the unified protocol while preserving its core verification idea, such as label consistency, boundary fingerprinting, multi-sample behavioral comparison, behavior-change consistency, adversarial trajectories, or boundary-distance-controlled fingerprinting.

Table 7. Overview of comparison methods under the unified evaluation protocol.

Method	Query Samples	Similarity Metric
Boundary-label	Adaptive FGSM samples	Label consistency only
IPGuard-like	Boundary samples	Boundary-label consistency
DeepJudge-like	Multi-type test samples	Averaged behavioral similarity
ModelDiff-like	Original–boundary input pairs	Output-change behavior similarity
ADV-TRA adaptation	Adversarial trajectory samples	Mutation-based trajectory consistency
AnaFP-like	Boundary-distance-controlled samples	Boundary-behavior similarity
Ours	Adaptive boundary fingerprints	Multi-metric fusion of S_{label}, S_{prob}, and S_{trans}

7. Experimental Results and Analysis

This chapter analyzes the effectiveness of the adaptive boundary fingerprint method from four aspects: model detection performance, comparison with baseline methods, ablation results, and parameter sensitivity. Baseline comparisons, ablations, and sensitivity analyses are mainly performed on CIFAR-100, which is used for complete derivation model coverage experiments, while Tiny-ImageNet is used for representative generalization validation.

7.1. Model Detection Results and Analysis

7.1.1. Detection Results

Unless otherwise specified, the main detection results are reported under the full-output black-box setting, where the target model returns confidence scores or probability vectors. Under this setting, the combined score uses S_{label}, S_{prob}, and S_{trans}.

Figure 2 shows the detection results on both datasets. On CIFAR-100, our method detects Copy, FT-All, Prune, Quant, and same-architecture boundary-enhanced distillation models; on Tiny-ImageNet, our method detects representative derived models such as Copy, FT-All-10%, Prune-40, and Quant-8bit.

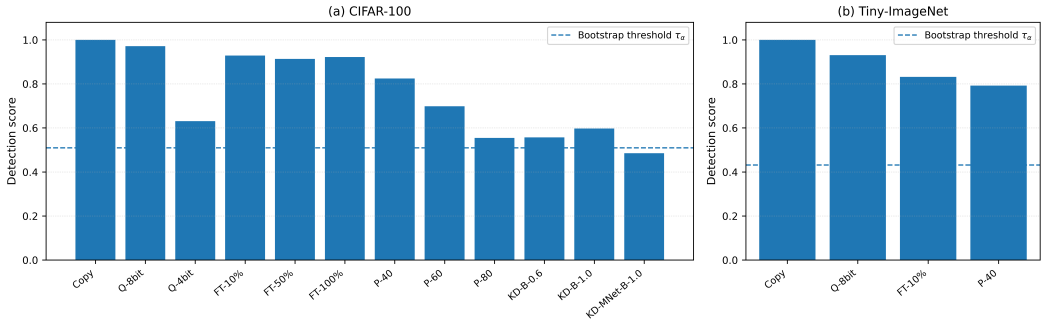


Figure 2. Main detection results on both datasets.

Table 8 shows the background distribution of independent models. On CIFAR-100, the detection scores of Copy and Quant-8bit are 1.000 and 0.971, respectively, substantially

higher than the formal threshold of 0.510. As the pruning ratio increases, the detection score decreases, but Prune-80 still yields a positive detection margin, indicating that strong pruning still leaves detectable source-model boundary behavior. On Tiny-ImageNet, representative derived models all exceed the threshold, showing that our method does not rely on a single dataset.

Table 8. Background distribution of independent models.

Dataset	Number of Independent Models	Mean Score	Max Score	Threshold	FPR
CIFAR-100	30	0.481	0.501	0.510	0.000
Tiny-ImageNet	30	0.384	0.402	0.431	0.000

The background distribution for threshold calibration is constructed from 30 independently trained ResNet18 models. Since these models share the same architecture and task setting as the source model but are trained with independent initialization and data order, they provide a representative same-architecture background for estimating the behavioral similarity that may arise without model derivation. Therefore, the calibrated threshold is based on independent models that are structurally close to the source model, which helps distinguish true model derivation from ordinary similarity among independently trained models.

Figure 3 shows the distribution of detection scores for independent and derived models. On both datasets, the scores of all 30 independent models are below the formal threshold, with a false-positive rate of 0. This result indicates that threshold calibration using the background distribution of independent models effectively controls false positives for non-derived models.

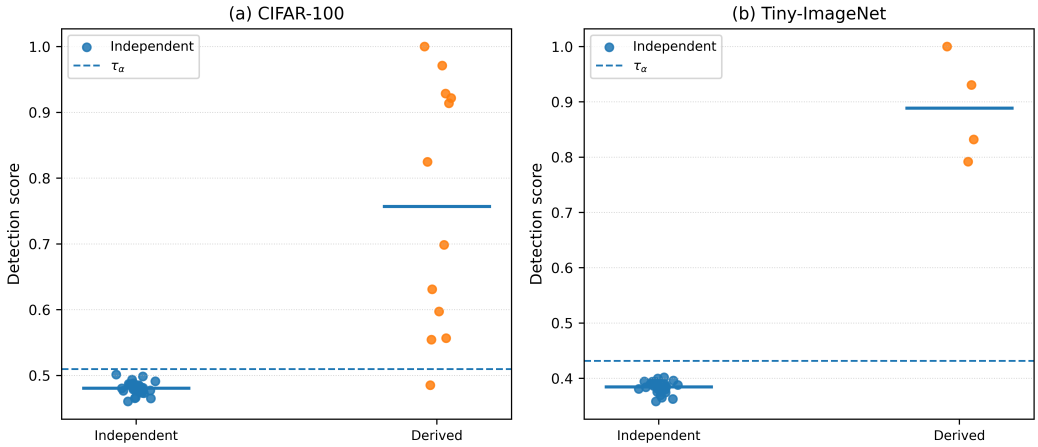


Figure 3. Distribution of detection scores for independent models and derived models.

7.1.2. Comparison with Baseline Methods

To compare different black-box copyright verification approaches, we conduct a multi-method comparison on CIFAR-100 under a unified evaluation protocol. Since existing fingerprinting methods were originally evaluated with different datasets, model sets, query-generation procedures, and threshold rules, the suffix "-like" denotes a functionally equivalent reimplementation that preserves the core mechanism of each method while using our source model, target model set, independent-model background distribution, query budget, and bootstrap threshold calibration. Since public code is available for ADV-TRA, we use an official-code-based adaptation, retaining its adversarial trajectory

generation and mutation-based verification logic while adapting model loading, data interface, and threshold calibration to our CIFAR-100 verification protocol.

Specifically, Boundary-label uses our adaptive boundary samples with label consistency only; IPGuard-like uses FGSM boundary samples with label consistency; DeepJudge-like averages behavioral similarity over Clean, Random, FGSM, and PGD samples; ModelDiff-like compares behavior changes between original-boundary input pairs; AnaFP-like uses boundary distance-controlled samples; and ADV-TRA adaptation uses official adversarial trajectories with mutation-based trajectory consistency. All methods are evaluated with approximately the same query budget.

As shown in Table 9, Boundary-label and ModelDiff-like detect 10 out of 12 derived models, indicating that using only label consistency or behavior-change consistency provides limited discriminative evidence. IPGuard-like and DeepJudge-like improve the detection rate to 11/12, but both produce false positives among independent models. AnaFP-like achieves the highest AUROC and AUPR and detects all derived models, showing strong threshold-independent separability, but it still yields 2/30 false positives under the calibrated fixed-threshold decision rule. The ADV-TRA official-code-based adaptation maintains zero false positives, but detects 10 out of 12 derived models and obtains lower AUROC and AUPR than our method. In comparison, our method detects 11 out of 12 derived models, keeps the false-positive rate at 0/30, and achieves AUROC and AUPR values of 0.9750 and 0.9643, respectively.

Table 9. Comparison with baseline methods on CIFAR-100 under the unified evaluation protocol.

Method	Verification Setting	Detection Rate	FPR	AUROC	AUPR
Boundary-label	Adaptive FGSM samples with label consistency	10/12	2/30	0.8903	0.8972
IPGuard-like	FGSM boundary samples with label consistency	11/12	2/30	0.9167	0.9405
DeepJudge-like	Clean, Random, FGSM, and PGD samples with averaged behavioral similarity	11/12	2/30	0.9167	0.9405
ModelDiff-like	Original-boundary input pairs with behavior-change consistency	10/12	0/30	0.8833	0.8953
AnaFP-like	Boundary distance-controlled samples with behavioral similarity	12/12	2/30	1.0000	1.0000
ADV-TRA adaptation	Official ADV-TRA trajectories with mutation-based trajectory consistency	10/12	0/30	0.9167	0.9405
Ours	Adaptive boundary fingerprints with multi-metric fusion	11/12	0/30	0.9750	0.9643

AnaFP-like achieves the highest AUROC and AUPR and detects all derived models, showing strong threshold-independent separability. However, its 2/30 false positives under the calibrated fixed-threshold rule indicate that perfect ranking does not necessarily imply zero false positives under a preset threshold. In our unified protocol, the bootstrap threshold is estimated from the independent-model background distribution at $\alpha_0 = 0.05$, rather than optimized after observing both derived and independent scores. Thus, the threshold can fall below a small upper-tail portion of the AnaFP-like independent-model scores, even though another threshold exists that would separate the two groups perfectly.

These results show that adaptive boundary fingerprint selection combined with multi-metric behavioral fusion provides a more favorable balance among fixed-threshold detection, threshold-independent ranking, and calibrated false-positive control. Compared with the baseline methods, our method maintains zero false positives while achieving a competitive detection rate and strong AUROC/AUPR, indicating more stable black-box verification performance.

7.1.3. Analysis of Distillation Model Detection

Figure 4 shows the multi-baseline detection margin comparison on boundary-enhanced distillation models. KD-Boundary-0.6 and KD-Boundary-1.0 obtain positive detection margins under various methods, indicating that when boundary-perturbed samples are used in distillation, the student model inherits the local boundary behavior of the teacher model.

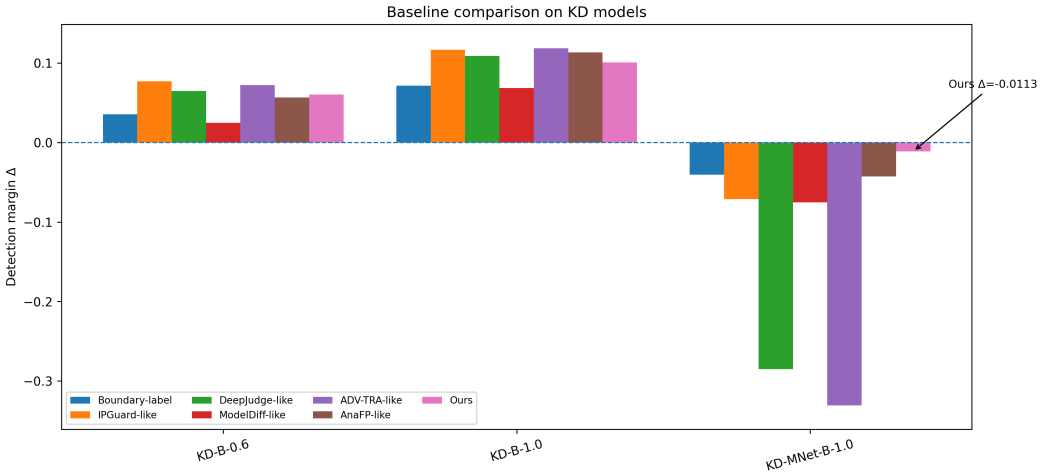


Figure 4. Multi-baseline detection margin comparison on boundary-enhanced distillation models.

Cross-architecture KD-MobileNet-Boundary-1.0 is the most challenging scenario. Although our method does not exceed the decision threshold, its detection margin is closest to zero, and it is closer to the positive decision region than DeepJudge-like, IPGuard-like, and ModelDiff-like, indicating that our method can still capture some boundary similarity signals. This result also shows that cross-architecture distillation markedly weakens the inheritance of the source model's local boundary behavior, which is an important limitation of boundary fingerprint detection. This is because the proposed fingerprints are constructed around the local decision boundary of the ResNet18 source model, whereas a MobileNetV2 student has different feature extraction structures and decision surface parameterization, leading to different local boundary geometry even when the global prediction behavior is similar. Distillation can preserve coarse prediction behavior on natural and boundary-perturbed samples, but it does not necessarily preserve the fine-grained local boundary geometry of the teacher model. Consequently, the label consistency, probability-distribution similarity, and boundary transition consistency on source-boundary fingerprints may all decrease, causing KD-MobileNet-Boundary-1.0 to remain below the calibrated threshold.

Figure 5 presents the detection results for boundary-enhanced distillation models from the perspectives of detection score and clean accuracy. KD-Boundary-1.0 has a higher detection score than KD-Boundary-0.6, indicating that more abundant in-distribution auxiliary data enhances boundary-behavior inheritance. Although KD-MobileNet-Boundary-1.0 still has reasonable task performance, its detection score does not exceed the formal threshold, indicating that structural differences weaken the consistency of local boundary behavior.

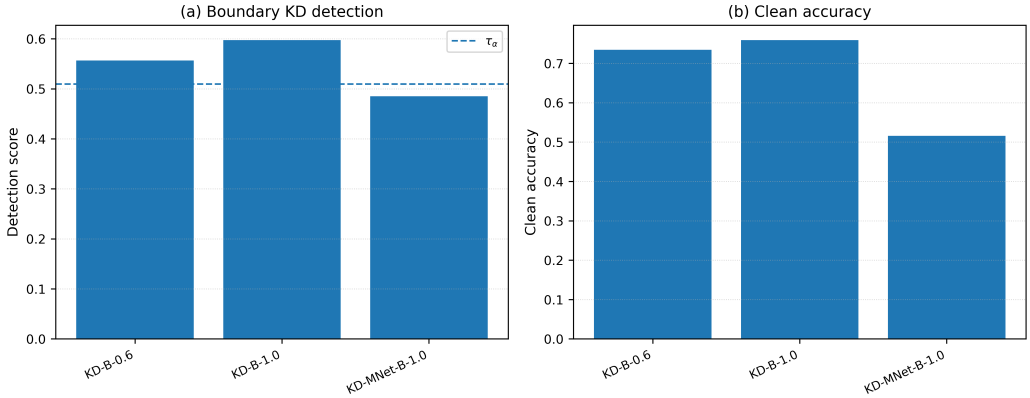


Figure 5. Detection results for boundary-enhanced distillation models.

7.2. Ablation Results and Analysis

7.2.1. Ablation on Query Sample Types

Figure 6 compares five types of query samples: Clean, Random, FGSM, PGD, and Adaptive FGSM. Clean and Random mainly reflect ordinary output consistency and cannot stably amplify boundary differences between models; FGSM and PGD introduce gradient-guided perturbations but lack boundary and diversity screening. Adaptive FGSM, by applying correctness, margin, class balance, and diversity constraints, maintains a more stable positive detection margin on most derived models.

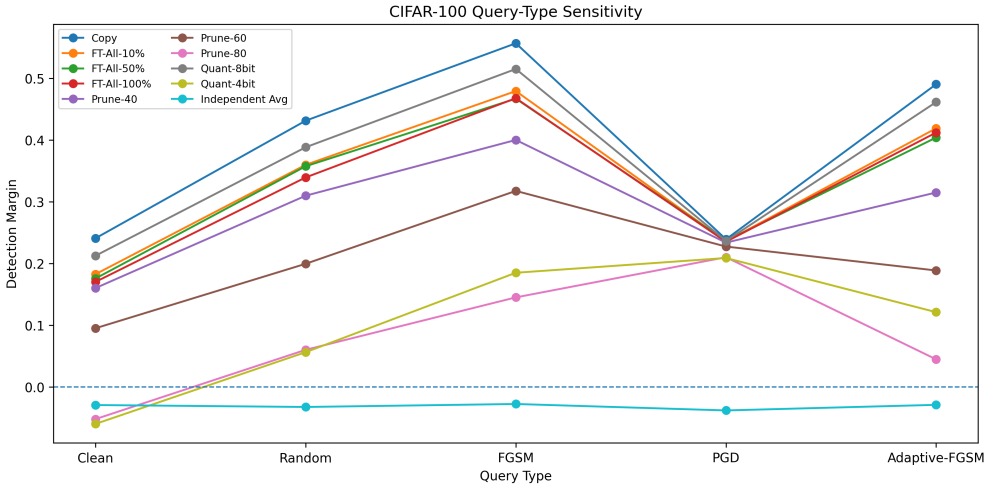


Figure 6. Impact of query sample type on detection margin.

7.2.2. Ablation and Sensitivity Analysis of Similarity Metrics

To jointly analyze the contribution of different similarity metrics and the effect of the score weights α, β, and γ, we evaluate both single-metric variants and representative weight combinations. For each setting, the detection score is recomputed using the same fingerprint set and the same component scores, and the decision threshold is calibrated using the same independent-model bootstrap procedure.

As shown in Table 10, the single-metric variants capture different aspects of model similarity. Prob-only achieves the strongest performance, detecting all 12 derived models with a false-positive rate of 0/30 and obtaining AUROC and AUPR values of 1.0000. This shows that probability-distribution similarity provides highly discriminative evidence when probability outputs are available. In comparison, Label-only and Trans-only both detect 10 out of 12 derived models with zero false positives, but obtain lower AUROC and

AUPR values, indicating that label consistency and boundary transition consistency are less discriminative in the full-output setting.

Table 10. Ablation and sensitivity analysis of similarity metrics and score weights.

Score Setting	α, β, γ	Detection Rate	FPR	AUROC	AUPR
Label-only	$1, 0, 0$	10/12	0/30	0.9056	0.9183
Prob-only	$0, 1, 0$	12/12	0/30	1.0000	1.0000
Trans-only	$0, 0, 1$	10/12	0/30	0.8833	0.8938
Equal	$1/3, 1/3, 1/3$	11/12	0/30	0.9750	0.9643
Label-heavy	$0.6, 0.2, 0.2$	10/12	0/30	0.9139	0.9335
Prob-heavy	$0.2, 0.6, 0.2$	11/12	0/30	0.9889	0.9792
Trans-heavy	$0.2, 0.2, 0.6$	10/12	0/30	0.9111	0.9277
Label+Prob	$0.5, 0.5, 0$	11/12	0/30	0.9722	0.9621
Label+Trans	$0.5, 0, 0.5$	10/12	0/30	0.8958	0.9054
Prob+Trans	$0, 0.5, 0.5$	11/12	0/30	0.9194	0.9411

The weight sensitivity results further show that all settings maintain a false-positive rate of 0/30. Among the multi-metric combinations, Prob-heavy achieves the highest AUROC and AUPR, confirming the importance of probability-level behavioral information. However, Prob-only is not used as the default setting because it depends entirely on probability-vector outputs and is not applicable to hard-label-only APIs. In addition, relying on a single metric may make the decision overly dependent on one type of behavioral evidence. Therefore, we use the Equal setting as a simple task-agnostic default, which combines label consistency, probability-distribution similarity, and boundary transition consistency without task-specific weight tuning.

7.2.3. Evaluation Under Input-Level Defense Mechanisms

Since the proposed method constructs candidate boundary fingerprints using gradient-based perturbations, it is necessary to examine whether input-level defense mechanisms weaken the effectiveness of boundary fingerprints. We therefore evaluate the target models under feature squeezing, input smoothing, and their combination. Feature squeezing reduces the input precision to 5 bits, while input smoothing applies a 3×3 smoothing operation before model inference. These settings are used to simulate preprocessing-based defenses that may suppress perturbation-sensitive responses around the decision boundary.

As shown in Table 11, feature squeezing alone does not weaken the verification result, and the proposed method still detects 11 out of 12 derived models with zero false positives. This indicates that simple input-precision reduction does not substantially change the boundary-related responses captured by the fingerprints. In contrast, when the combined score is used, input smoothing significantly reduces the detection rate, showing that smoothing operations can suppress local perturbation-sensitive label and transition behaviors near the decision boundary.

Table 11. Verification results under input-level defense mechanisms.

Defense Setting	Score Setting	Detection Rate	FPR	AUROC	AUPR
Feature squeezing 5-bit	Combined	11/12	0/30	0.9167	0.9405
Input smoothing (k = 3)	Combined	1/12	0/30	0.6917	0.4930
Input smoothing (k = 3)	Prob-only	11/12	0/30	0.9806	0.9693
Smoothing + squeezing	Combined	1/12	0/30	0.8139	0.7148
Smoothing + squeezing	Prob-only	12/12	0/30	1.0000	1.0000

However, the results under the Prob-only setting further show that probability-distribution similarity remains highly discriminative when probability outputs are avail-

able. Under input smoothing, Prob-only still detects 11 out of 12 derived models with zero false positives, and under the combined smoothing and squeezing setting, it detects all 12 derived models with zero false positives. These results suggest that input smoothing mainly weakens hard-label and boundary-transition signals, while probability-level behavioral similarity is more stable under such preprocessing defenses. Therefore, the proposed multi-level verification framework can still benefit from probability outputs in smoothed target-model settings, although robust verification under hard-label-only smoothing defenses remains a challenging case.

7.3. Sensitivity Results and Analysis

This section analyzes the impact of fingerprint sample size, perturbation strength, classification margin threshold, and restricted hard-label-only output settings on detection performance. Since CIFAR-100 covers a more complete set of derived model types, sensitivity analyses are mainly presented on CIFAR-100.

7.3.1. Impact of Fingerprint Sample Size

Figure 7 shows the effect of fingerprint sample size on detection margin. When the fingerprint size increases from 50 to 500, the detection margin of most derived models stabilizes, indicating that increasing the query budget reduces fluctuations caused by randomness in individual samples. Continuing to increase to 1000 yields limited additional benefit, so we default to $K = 500$ as a trade-off between query cost and detection stability.

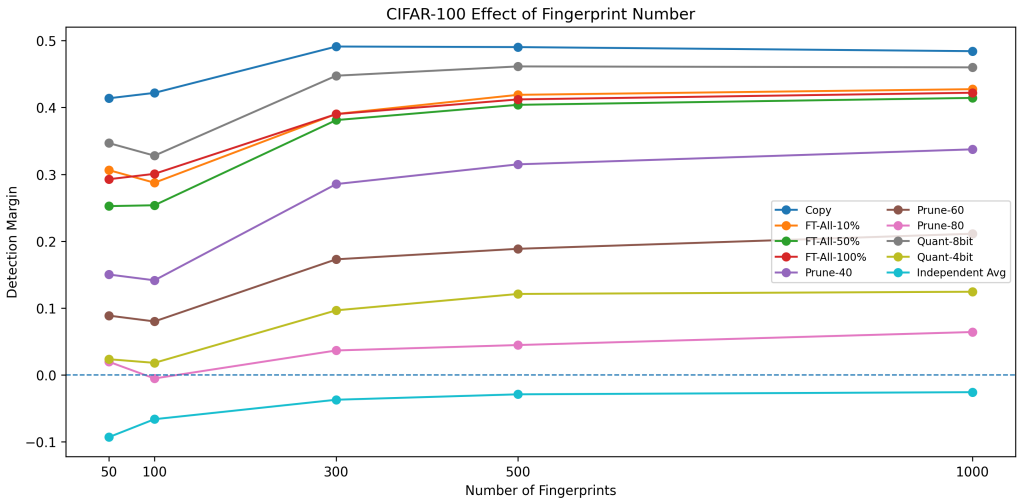


Figure 7. Impact of fingerprint sample size on detection margin.

7.3.2. Impact of Perturbation Strength

Figure 8 shows the effect of perturbation strength ϵ on detection margin. If the perturbation strength is too small, samples do not sufficiently approach the source model's decision boundary, limiting the amplification of boundary differences; if the perturbation strength is too large, samples may deviate from the original semantics or reduce the quality of usable boundary samples after adaptive filtering. In terms of efficiency, the per-sample cost of FGSM remains one gradient computation for different ϵ, but an unsuitable ϵ may reduce the number of valid candidates after adaptive filtering and therefore require a larger candidate pool. The default setting of 8/255 achieves a good balance between detection stability, candidate availability, and sample effectiveness.

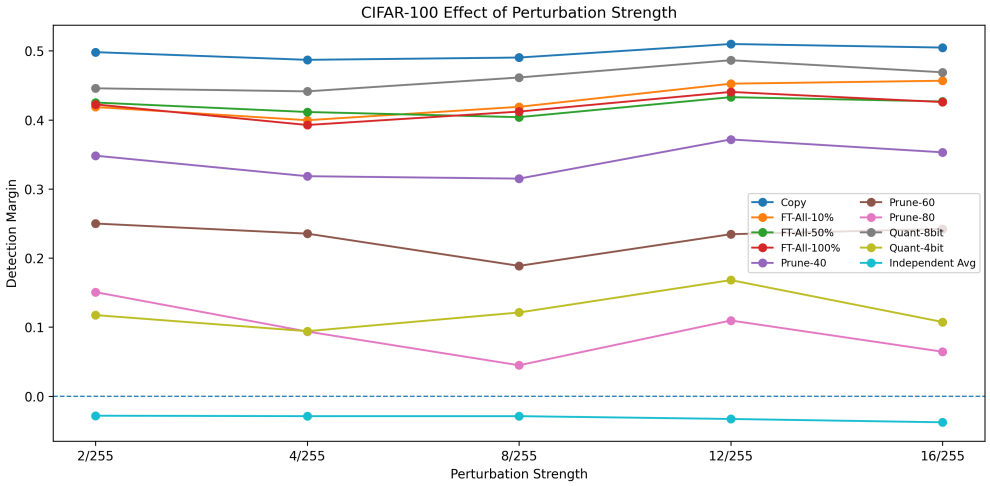


Figure 8. Impact of perturbation strength on detection margin.

7.3.3. Impact of Classification Margin Threshold

Figure 9 shows the effect of classification margin threshold on detection margin. The margin threshold controls the strictness of boundary screening. A smaller margin threshold selects samples that are closer to the source model's decision boundary, but may lead to insufficient candidate samples, reduced class coverage, and lower sample diversity. A larger margin threshold relaxes the screening conditions, allowing more samples into the fingerprint set, but may also introduce samples with insufficient boundary proximity. Experimental results show that the default threshold of 1.0 provides a good trade-off between sample boundary proximity and sample availability.

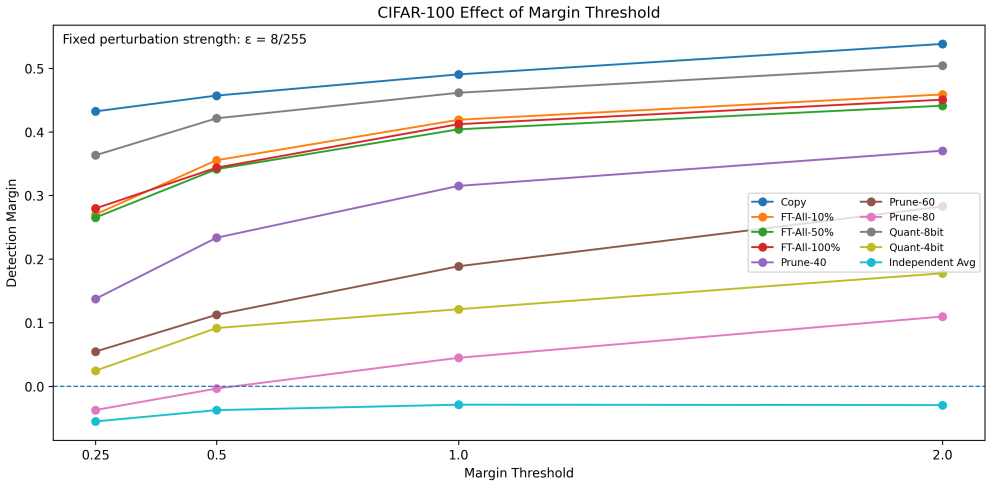


Figure 9. Impact of classification margin threshold on detection margin.

In summary, the experimental results demonstrate that our method can stably detect parameter-inherited derived models under black-box conditions, and can recognize boundary-behavior inheritance from same-architecture boundary-enhanced distillation. Compared with single-label consistency or single-boundary transition consistency metrics, the multi-metric fusion achieves more stable overall performance in fixed-threshold false-positive control and threshold-independent ranking ability.

7.3.4. Impact of Hard-Label-Only Outputs

To evaluate whether the proposed method remains applicable when the target API only returns hard labels, we further test several restricted-output settings. The main

detection results reported above correspond to the full-output setting, where confidence scores or probability vectors are available. In the restricted hard-label-only setting, S_{prob} is unavailable, and the detection score follows Equation (26). Specifically, we evaluate label consistency, boundary transition consistency, and their weighted combinations with different values of λ.

As shown in Table 12, the proposed framework can still operate under the hard-label-only setting. Compared with the full-output mode, the detection rate decreases from 11/12 to 10/12, and the AUROC decreases from 0.9750 to about 0.90. This confirms that probability-distribution similarity provides useful confidence-level evidence when available. Nevertheless, the hard-label variants still maintain a false-positive rate of 0/30 and detect 10 out of 12 derived models, showing that the verification mechanism is not strictly dependent on complete softmax outputs. Among the hard-label settings, label consistency and label-dominant combinations perform better than using boundary transition consistency alone, indicating that S_{trans} is more suitable as auxiliary hard-label evidence. Overall, complete probability outputs improve detection sensitivity, while the hard-label degradation strategy preserves the applicability of the method in more restrictive black-box API scenarios.

Table 12. Evaluation under hard-label-only black-box output settings.

Score Setting	λ	Detection Rate	FPR	AUROC	AUPR
S_{label}	1.0	10/12	0/30	0.9056	0.9183
S_{trans}	0.0	10/12	0/30	0.8833	0.8938
$S_{\text{label}} + S_{\text{trans}}$	0.3	10/12	0/30	0.8806	0.8953
$S_{\text{label}} + S_{\text{trans}}$	0.5	10/12	0/30	0.8958	0.9054
$S_{\text{label}} + S_{\text{trans}}$	0.7	10/12	0/30	0.9028	0.9144

8. Conclusions and Future Work

8.1. Conclusions

This paper proposes a black-box model copyright verification method based on adaptive boundary fingerprints. The method does not rely on the internal structure or parameters of the target model. Instead, it generates boundary fingerprint samples from the source model and determines whether the target model is suspected to be derived by comparing the output behavioral similarity between the source and target models on these samples. Compared with ordinary test samples, boundary fingerprint samples are closer to the local decision boundary of the source model, thereby more effectively exposing behavioral differences between derived models and independent models. Experimental results show that the proposed method can identify most evaluated model derivation methods, including direct copying, fine-tuning, pruning, quantization, and same-architecture boundary-enhanced distillation. At the same time, we perform quantile threshold calibration based on the background distribution of independent models and use the maximum-background score as a conservative reference, reducing the risk of false positives caused by natural similarity among a limited number of independent models. Ablation experiments also show that ordinary test samples easily yield high similarity scores for independent models, whereas FGSM and adaptive boundary fingerprint samples more effectively reduce the background scores of independent models, thus improving detection discrimination. Overall, the proposed method is suitable for black-box verification scenarios where the target model's parameters and structure are inaccessible, providing a technical reference for deep model copyright protection and infringement evidence collection.

8.2. Future Work

There is still room for further improvement of the proposed method. First, the threshold calibration relies on the background distribution of independent models; the representativeness of the set of background models affects the stability of the estimated similarity range for non-derived models. In the future, we can introduce more random seeds, model architectures, training strategies, data augmentation methods, and pre-training sources to build a richer independent-model background library, improving the robustness of threshold calibration. Second, fingerprint sample generation requires a trade-off between boundary proximity, naturalness, and query cost. Perturbations that are too small may not sufficiently approach the decision boundary, while perturbations that are too large may cause samples to deviate from the original data distribution, leading to detection results affected by out-of-distribution responses. Therefore, future work could investigate adaptive methods for setting perturbation strength, margin threshold, and sample selection strategies, reducing the impact of manual parameter choices on detection performance. Third, stronger evasion scenarios such as cross-architecture distillation, large-scale retraining, adversarial training, randomized smoothing, boundary smoothing, and output randomization may weaken the inheritance of the source model's boundary behavior. In particular, adversarial training may enlarge or reshape local classification margins, while randomized smoothing may smooth model responses around the decision boundary, both of which can reduce the effectiveness of boundary fingerprints constructed from the source model. In addition, the current experiments are mainly conducted on CIFAR-100, Tiny-ImageNet, and CNN-based architectures such as ResNet18 and MobileNetV2. Evaluation on larger datasets such as ImageNet and more diverse model configurations, such as ViT-B/16, Swin-T, and ConvNeXt-T, remains an important direction for future work. Future research could explore dynamic fingerprint generation, multi-round queries, and private fingerprint library update mechanisms to reduce the risk of fixed fingerprints being evaded in a targeted manner. Finally, black-box model copyright verification should not rely solely on a single type of technical evidence. In practice, boundary fingerprint methods can be combined with model watermarking, access log auditing, trusted timestamps, model registration, and training data provenance to form a more complete system for model copyright protection and infringement evidence collection.

Author Contributions: Conceptualization, X.Y. and Z.G.; methodology, X.Y.; software, X.Y.; validation, X.Y. and T.Z.; formal analysis, X.Y.; investigation, X.Y.; resources, Z.G.; data curation, X.Y.; writing—original draft preparation, X.Y.; writing—review and editing, Z.G. and T.Z.; visualization, X.Y.; supervision, Z.G.; project administration, Z.G.; funding acquisition, Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Beijing–Tianjin–Hebei Natural Science Foundation Cooperation Project, grant number 25JJJJC0037. And the Science and Technology Innovation Program of Hunan Province, grant number 2026JK2081.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available. CIFAR-100 and Tiny-ImageNet can be accessed from their official repositories. The experimental data generated in this study are available from the corresponding author upon reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT (GPT-5.5) only for language polishing and formatting assistance. The authors reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADV-TRA	Adversarial Trajectory
AI	Artificial Intelligence
API	Application Programming Interface
AUPR	Area Under the Precision-Recall Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
CIFAR-100	Canadian Institute for Advanced Research 100-class dataset
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DAWN	Dynamic Adversarial Watermarking of Neural Networks
DNN	Deep Neural Network
FGSM	Fast Gradient Sign Method
FPR	False-Positive Rate
FT	Fine-Tuning
GPU	Graphics Processing Unit
JS	Jensen-Shannon Divergence
KD	Knowledge Distillation
LLM	Large Language Model
LTS	Long-Term Support
PGD	Projected Gradient Descent
SDBF	Steep-Decision-Boundary Fingerprinting
Tiny-ImageNet	Tiny-ImageNet dataset

References

1. Sun, Y.; Liu, T.; Hu, P.; Liao, Q.; Fu, S.; Yu, N.; Guo, D.; Liu, Y.; Liu, L. Deep intellectual property protection: A survey. *arXiv* **2023**, arXiv:2304.14613.
2. Lederer, I.; Mayer, R.; Rauber, A. Identifying appropriate intellectual property protection mechanisms for machine learning models: A systematization of watermarking, fingerprinting, model access, and attacks. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 13082–13100. [CrossRef]
3. Chen, H.; Liu, C.; Zhu, T.; Zhou, W. When deep learning meets watermarking: A survey of application, attacks and defenses. *Comput. Stand. Interfaces* **2024**, *89*, 103830. [CrossRef]
4. Uchida, Y.; Nagai, Y.; Sakazawa, S.; Satoh, S.I. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*; ACM: New York, NY, USA, 2017; pp. 269–277.
5. Zhang, J.; Gu, Z.; Jang, J.; Wu, H.; Stoecklin, M.P.; Huang, H.; Molloy, I. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*; ACM: New York, NY, USA, 2018; pp. 159–172.
6. Adi, Y.; Baum, C.; Cisse, M.; Pinkas, B.; Keshet, J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security 18)*; USENIX Association: Berkeley, CA, USA, 2018; pp. 1615–1631.
7. Darvish Rouhani, B.; Chen, H.; Koushanfar, F. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*; ACM: New York, NY, USA, 2019; pp. 485–497.
8. Chen, H.; Rouhani, B.D.; Koushanfar, F. Blackmarks: Blackbox multibit watermarking for deep neural networks. *arXiv* **2019**, arXiv:1904.00344.
9. Szyller, S.; Atli, B.G.; Marchal, S.; Asokan, N. Dawn: Dynamic adversarial watermarking of neural networks. In *Proceedings of the 29th ACM International Conference on Multimedia*; ACM: New York, NY, USA, 2021; pp. 4417–4425.
10. Lukas, N.; Jiang, E.; Li, X.; Kerschbaum, F. Sok: How robust is image classification deep neural network watermarking? In *Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP)*; IEEE: New York, NY, USA, 2022; pp. 787–804.
11. Shao, S.; Li, Y.; Yao, H.; He, Y.; Qin, Z.; Ren, K. Explanation as a watermark: Towards harmless and multi-bit model ownership verification via watermarking feature attribution. *arXiv* **2024**, arXiv:2405.04825.

12. Krauß, T.; Stang, J.; Dmitrienko, A. ClearStamp: A Human-Visible and Robust Model-Ownership Proof based on Transposed Model Training. In *Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24)*; USENIX Association: Berkeley, CA, USA, 2024; pp. 5269–5286.
13. Pautov, M.; Bogdanov, N.; Pyatkin, S.; Rogov, O.; Oseledets, I. Probabilistically robust watermarking of neural networks. *arXiv* **2024**, arXiv:2401.08261.
14. Yan, Y.; Pan, X.; Zhang, M.; Yang, M. Rethinking White-Box watermarks on deep learning models under neural structural obfuscation. In *Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23)*; USENIX Association: Berkeley, CA, USA, 2023; pp. 2347–2364.
15. Chen, Y.; Tian, J.; Chen, X.; Zhou, J. Effective ambiguity attack against passport-based dnn intellectual property protection schemes through fully connected layer substitution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2023; pp. 8123–8132.
16. Pan, L.; Liu, A.; Huang, S.; Lu, Y.; Hu, X.; Wen, L.; King, I.; Yu, P.S. Can LLM Watermarks Robustly Prevent Unauthorized Knowledge Distillation? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 13228–13251.
17. Cao, X.; Jia, J.; Gong, N.Z. 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*; ACM: New York, NY, USA, 2021; pp. 14–25.
18. Chen, J.; Wang, J.; Peng, T.; Sun, Y.; Cheng, P.; Ji, S.; Ma, X.; Li, B.; Song, D. Copy, right? a testing framework for copyright protection of deep learning models. In *Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP)*; IEEE: New York, NY, USA, 2022; pp. 824–841.
19. Li, Y.; Zhang, Z.; Liu, B.; Yang, Z.; Liu, Y. ModelDiff: Testing-based DNN similarity comparison for model reuse detection. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*; ACM: New York, NY, USA, 2021; pp. 139–151.
20. Xu, T.; Wang, C.; Liu, G.; Yang, Y.; Peng, K.; Liu, W. United we stand, divided we fall: Fingerprinting deep neural networks via adversarial trajectories. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 69299–69328. [CrossRef]
21. Yang, G.; Geng, Z.; Chen, Y.; Luo, C. Fingerprinting deep neural networks for ownership protection: An analytical approach. *arXiv* **2026**, arXiv:2603.21411.
22. Sun, H.; Zhang, L.; Li, X.Y. DeepCore: Simple Fingerprint Construction for Differentiating Homologous and Piracy Models. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer Nature: Cham, Switzerland, 2025; pp. 328–344.
23. Bai, X.; Li, S.; Ma, X.; Zhu, B.B.; Zhang, D.; Yu, L. SDBF: Steep-Decision-Boundary Fingerprinting for Hard-Label Tampering Detection of DNN Models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*; IEEE: New York, NY, USA, 2025; pp. 29278–29287.
24. Yamabe, S.; Waseda, F.K.; Takahashi, T.; Wataoka, K. 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)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 6894–6916.
25. Hu, A.; Lu, Z.; Xie, R.; Xue, M. VeriDIP: Verifying Ownership of Deep Neural Networks Through Privacy Leakage Fingerprints. *IEEE Trans. Dependable Secur. Comput.* **2023**, *21*, 2568–2584. [CrossRef]
26. Liu, W.; Zhong, S. Marginfinger: Controlling generated fingerprint distance to classification boundary using conditional gans. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*; ACM: New York, NY, USA, 2024; pp. 129–136.
27. Mo, M.; Wang, C.; Guo, Q.; Bian, S.; Huang, Q.; Zhang, X. A novel robust black-box fingerprinting scheme for deep classification neural networks. *Expert Syst. Appl.* **2024**, *252*, 124201. [CrossRef]
28. Ren, H.; Yan, A.; Ren, X.; Ye, P.G.; Gao, C.Z.; Zhou, Z.; Li, J. Ganfinger: Gan-based fingerprint generation for deep neural network ownership verification. *arXiv* **2023**, arXiv:2312.15617.
29. Tang, L.; Chen, Y.F.; Xue, H.; Zhang, Q. Towards the resistance of neural network fingerprinting to fine-tuning. *Adv. Neural Inf. Process. Syst.* **2026**, *38*, 147167–147193.
30. Liu, J.; Zhang, R.; Szyller, S.; Ren, K.; Asokan, N. False claims against model ownership resolution. In *Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24)*; USENIX Association: Berkeley, CA, USA, 2024; pp. 6885–6902.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.