REVEAL: FINGERPRINTING TO DETECT MODEL REUSE IN BLACK-BOX ENSEMBLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Unauthorized reuse of a proprietary deep neural network (DNN) or its derived variants can be concealed within an ensemble, where prediction aggregation obscures individual model behavior. Existing fingerprinting methods primarily target standalone models, and their ownership-indicative signals can become substantially less distinguishable after aggregation. To address this challenge, we propose REVEAL, a model fingerprinting framework for detecting model **RE**use in black-box ensembles **V**ia fingerprints generated through **E**nsemble-**A**ware **L**earning. Specifically, REVEAL derives fixed verification targets from naturally occurring prediction errors of the protected model and optimizes fingerprint queries without modifying model parameters. Its key idea is to reinforce the protected model's target responses while eliciting aggregation-neutral responses from independently trained reference models, which serve as surrogates for unrelated ensemble members. For single-label classification, reference predictions are driven toward uniformity to limit competing class preferences. For multi-label classification, reference probabilities for the selected target labels are driven slightly below the decision threshold, allowing a strong protected-model response to influence the final label decision. Ownership verification requires only the ensemble's returned labels and aggregates matches with the expected fingerprint targets into an ownership score. Extensive experiments demonstrate that REVEAL substantially mitigates the verification degradation caused by ensemble aggregation and effectively detects reuse of both protected models and their derived variants.

1 INTRODUCTION

Model ensembles are a well-established learning paradigm that combines multiple deep neural networks (DNNs) into a unified predictive system, rather than relying on a single model (Dietterich, 2000; Lakshminarayanan et al., 2017; Huang et al., 2017; Ganaie et al., 2022). However, the flexibility that makes ensembles attractive also introduces an intellectual-property risk: an adversary may incorporate a proprietary DNN model, either directly or after modification, as an ensemble member without authorization. Since only the ensemble's aggregated output is exposed, the individual behavior of the reused model is concealed, making such unauthorized reuse difficult to verify.

Model fingerprinting has emerged as a promising, non-intrusive approach for verifying model ownership without modifying the protected model (Cao et al., 2021; Peng et al., 2022; Xu et al., 2024; Godinot et al., 2025; Zhang et al., 2025; Geng et al., 2026). However, existing fingerprinting approaches are designed primarily for standalone model verification (Lukas et al., 2021; Pan et al., 2022; Liu & Zhong, 2024; Xu et al., 2026b; Yang et al., 2026a; Xu et al., 2026a). These approaches construct fingerprint queries that elicit behavioral characteristics shared by the protected model and its derived variants while remaining distinguishable from independently trained models. Ownership is then verified by directly comparing the suspect model's responses with those induced by the protected model.

However, this standalone verification paradigm does not readily extend to detecting model reuse within ensembles. A conventional ensemble combines models that perform the same prediction task but differ in architecture, initialization, or training, averaging their output probability vectors equally to determine the final prediction. For each fingerprint query, the verifier observes only this final prediction, jointly determined by the reused model and the other ensemble members. Contribu-

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

tions from these other members can attenuate or obscure the protected model's ownership-indicative signal, making its distinctive behavior less distinguishable in the observable output. Consequently, fingerprints that reliably identify a protected model or its derived variants under standalone deployment may lose substantial verification capability when the model is incorporated into an ensemble.

To empirically assess this limitation, we evaluate existing model fingerprinting approaches across two datasets, with the results summarized in Fig. 1. As shown, fingerprints, produced by these existing approaches, provide clear ownership evidence when the protected model is deployed standalone, but their verification performance degrades substantially when the model is incorporated into an ensemble. This degradation demonstrates that ensemble aggregation weakens the ownership-indicative signals captured by fingerprints generated by existing fingerprinting approaches, motivating a fingerprinting approach specifically designed for ensemble-based ownership verification.

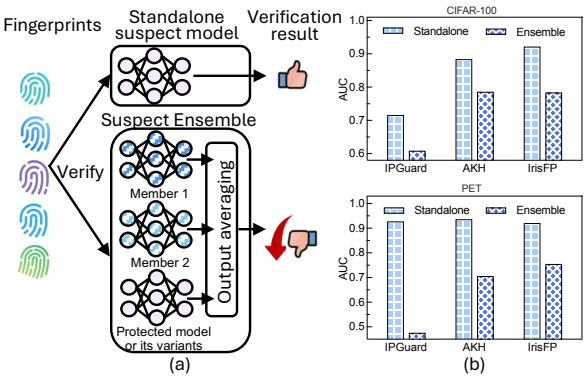

Figure 1: (a) Standalone and ensemble-based ownership verification scenarios; (b) The AUCs achieved by existing fingerprint approaches under standalone and ensemble deployment.

Main research question: *Can we design a model fingerprinting method that reliably verifies whether a black-box ensemble contains a protected model or one of its derived variants, using only the ensemble's aggregated predictions?*

Our Answer: In this paper, we provide an affirmative answer to the above question. Our key insight is that equal-weight probability averaging can preserve ownership evidence when independently trained, unrelated ensemble members produce aggregation-neutral responses. These responses introduce little competing bias, helping the protected model's target-specific behavior remain distinguishable in the final ensemble prediction. Therefore, we optimize fingerprint queries to jointly reinforce the protected model's target responses and shape independently trained reference models toward aggregation-neutral responses tailored to the task's prediction mechanism. The reference models serve as surrogates for unknown ensemble members. This design aims to preserve recognized ownership evidence when the protected model or one of its derived variants is incorporated into an ensemble.

The desired aggregation-neutral response depends on the classification setting. For single-label classification, a near-uniform probability vector contributes approximately equally to all classes, introducing little preference for competing classes under averaging. For multi-label classification, each label's averaged probability is independently compared with a decision threshold. Keeping unrelated members' probabilities for the fingerprint target slightly below this threshold prevents an ensemble composed solely of such members from predicting the target as present. Meanwhile, a sufficiently strong response from a reused protected model can raise the averaged probability above the threshold, preserving the target label in the ensemble's final prediction.

Our Design: We propose REVEAL, a model fingerprinting framework for detecting model **RE**use in black-box ensembles **V**ia fingerprints generated through **E**nsemble-**A**ware **L**earning. REVEAL supports both single-label and multi-label classification. It comprises two processes: fingerprint generation and ownership verification.

During fingerprint generation, REVEAL initializes fingerprint queries from samples that expose naturally occurring prediction errors of the protected model and retains an erroneous predicted label as each query's target, inspired by the initialization method from Godinot et al. (2025). Since the actual ensemble members are unknown, REVEAL utilizes independently trained reference models as surrogates for unrelated members. It jointly optimizes the queries under two complementary objectives: reinforcing the protected model's target responses and shaping the reference models' responses to limit interference under ensemble averaging. All model parameters and target labels are fixed, while reference models are randomly sampled throughout optimization to improve generalization to un-

seen ensemble members. The optimized queries and their fixed targets form the final fingerprint set.

During ownership verification, the verifier submits fingerprint queries to a suspect ensemble and checks whether each returned label set contains the corresponding target label. These matching outcomes are aggregated into an ownership score, with higher scores indicating stronger evidence that the ensemble reuses the protected model or one of its derived variants. This verification rule applies to both single-label and multi-label classification and relies solely on the ensemble's returned labels.

2 Problem Formulation

We focus on model ensembles in which all members are DNNs that perform the same prediction task over a shared label space $\mathcal{C} = \{1, \ldots, C\}$ with C classes. Given an input, each member produces a C-dimensional probability vector, and the ensemble averages these vectors with equal weights. We consider two task settings: single-label classification, which selects the class with the highest aggregated probability, and multi-label classification, which independently determines the presence of each label by comparing its aggregated probability against a predefined decision threshold.

We adopt a typical model fingerprinting scenario involving two parties: a model owner, who also acts as the verifier, and an adversary. The owner possesses a proprietary model P, referred to as the protected model, trained on a dataset $\mathcal{D}$ defined over the shared label space. The adversary steals P and incorporates either P itself or a derived variant P' into an ensemble as a member. Such variants are obtained through commonly studied model modification techniques, including fine-tuning, pruning, knowledge distillation, or their combinations. To support ownership verification, the owner constructs a fingerprint set $\mathcal{F} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{N}$ for the protected model P, where $\hat{x}_i$ is a fingerprint query and $\hat{y}_i$ is the corresponding target label. The owner keeps $\mathcal{F}$ confidential and utilizes it to detect unauthorized reuse of P or one of its derived variants within a suspect ensemble.

2.1 Unauthorized Model Reuse in Model Ensembles

The adversary incorporates either P or a derived variant P' into an ensemble $M = \{E_1, \ldots, E_K\}$ without authorization. Among its K members, E_{j^*} denotes the reused model, while the rest $K-1$ members are independently trained and unrelated to P. Given an input x, the prediction rules for both individual members and the ensemble depend on the task setting: single-label or multi-label classification.

Single-label classification. For single-label classification, each member E_j of the ensemble M produces a probability vector $p_j(x) \in (0,1)^C$ for an input x, satisfying $\sum_{c=1}^{C}[p_j(x)]_c = 1$. Here, $[p_j(x)]_c$ denotes the probability assigned to class c by E_j. The ensemble computes the equally weighted average $p_s(x) = \frac{1}{K}\sum_{j=1}^{K} p_j(x)$ and selects the class with the highest aggregated probability. To unify the output representation across single-label and multi-label classification, we express the ensemble's single-label prediction as a singleton label set: $\tilde{y}_s(x) = \{\arg\max_{c\in\mathcal{C}}[p_s(x)]_c\}$, where $[p_s(x)]_c$ is the ensemble's aggregated probability for class c.

Multi-label classification. For multi-label classification, each member E_j of the ensemble M produces a probability vector $p_j(x) \in (0,1)^C$ for an input x, where $[p_j(x)]_c$ denotes the predicted probability that label c is present. Unlike the single-label setting, these probabilities need not sum to one, since multiple labels may be present simultaneously. The ensemble computes the equally weighted average $p_s(x) = \frac{1}{K}\sum_{j=1}^{K} p_j(x)$ and independently compares each label's aggregated probability with a common decision threshold τ. Then, it returns the set of labels predicted as present: $\tilde{y}_s(x) = \{c \mid [p_s(x)]_c \geq \tau, c \in \mathcal{C}\}$.

Design goals. We consider black-box verification: the verifier observes only the predicted labels returned by a suspect ensemble M_s. Neither the aggregated probability vector nor the probability vectors of individual members are accessible. Our goal is to develop a fingerprinting approach that enables the verifier to determine, solely from these predicted labels, whether M_s contains the protected model P or a derived variant P', in both single-label and multi-label classification settings.

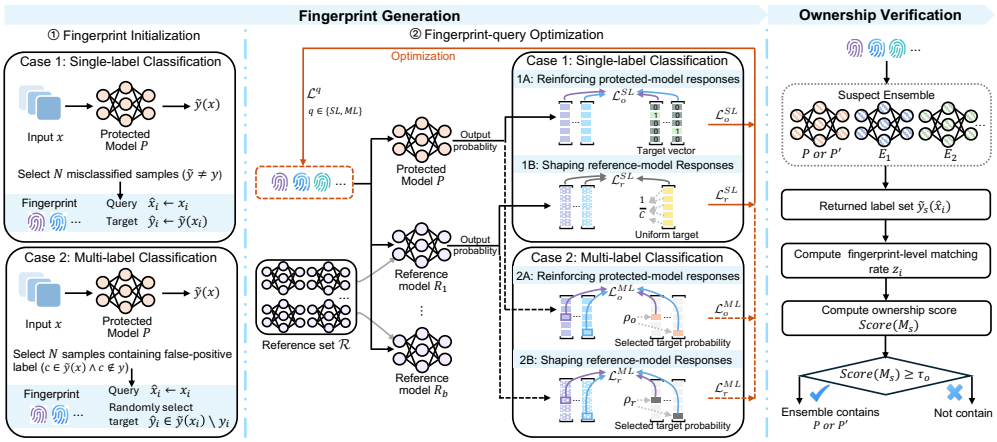

Figure 2: The overview of REVEAL.

3 METHODOLOGY

3.1 OVERVIEW

We present REVEAL, a model fingerprinting framework for detecting unauthorized model reuse in black-box ensembles across both single-label and multi-label classification tasks. REVEAL utilizes aggregation-aware fingerprint learning to reinforce the protected model's distinctive responses while limiting interference from independently trained, unrelated ensemble members. As demonstrated in Fig. 2, REVEAL consists of two main processes: fingerprint generation and ownership verification.

During fingerprint generation, REVEAL selects samples that expose naturally occurring prediction errors of the protected model. Each selected sample initializes a fingerprint query, and an erroneous label predicted by the protected model is retained as its target. Since the actual ensemble members are unknown during fingerprint generation, REVEAL uses a reference set of independently trained models as surrogates for unrelated ensemble members. Then, queries are optimized under two complementary objectives: reinforcing the protected model's response to each query's target label and shaping the reference models' responses to limit interference under ensemble averaging. The target labels and all model parameters remain fixed throughout optimization. Reference models are randomly sampled from the reference set throughout optimization to improve generalization to unseen ensemble members. The optimized queries and their fixed target labels form the final fingerprint set.

During ownership verification, the verifier submits the fingerprint queries to a suspect ensemble through its black-box interface and checks whether each returned label set contains the corresponding target label. These matching outcomes across all queries are aggregated into an ownership score, with higher scores indicating stronger evidence that the ensemble contains the protected model or one of its derived variants.

3.2 FINGERPRINT GENERATION

Given a protected model P trained on a dataset $\mathcal{D}$ with C classes, REVEAL generates fingerprints in two steps: initialization from natural prediction errors and query optimization. We describe each step below for both single-label and multi-label classification.

3.2.1 FINGERPRINT INITIALIZATION

Case 1: Single-label Classification. Consider a sample $(x, y) \in \mathcal{D}$, where x is the input and $y \in \mathcal{C} = \{1, \ldots, C\}$ is its ground-truth label. Let $p_o(x)$ denote the protected model's predictive probability vector and $\tilde{y}(x) = \{\arg\max_{c \in \mathcal{C}} [p_o(x)]_c\}$ its predicted class. The sample is naturally misclassified when $\tilde{y}(x) \neq y$. From these misclassified samples, we select N such samples $\{(x_i, y_i)\}_{i=1}^{N}$. For each selected input, we initialize a fingerprint query as $\hat{x}_i \leftarrow x_i$ and retain the protected model's erroneous prediction as the fingerprint target, i.e., $\hat{y}_i \leftarrow \tilde{y}(x_i)$. The target $\hat{y}_i$ remains fixed throughout subsequent optimization of the corresponding query $\hat{x}_i$.

Case 2: Multi-label Classification. Consider a sample $(x, y) \in \mathcal{D}$, where x is the input and $y \subseteq \mathcal{C}$ is its ground-truth label set. Let $p_o(x)$ denote the protected model's predictive probability vector, and $\tilde{y}(x) = \{c \mid [p_o(x)]_c \geq \tau\}$ its predicted label set under the decision rule defined in Sec. 2. A sample is eligible for fingerprint initialization if P predicts at least one false-positive label, i.e., $\exists c \in \mathcal{C} : c \in \tilde{y}(x) \wedge c \notin y$. We randomly select N such samples $\{(x_i, y_i)\}_{i=1}^{N}$. For each selected sample, we initialize a fingerprint query as $\hat{x}_i \leftarrow x_i$ and randomly choose one target label from its false-positive set, i.e., $\hat{y}_i \in \tilde{y}(x_i) \setminus y_i$. The target remains fixed throughout subsequent query optimization, which explicitly shapes only the model responses corresponding to $\hat{y}_i$, leaving the remaining label dimensions unconstrained.

3.2.2 Fingerprint-Query Optimization

To realize REVEAL's design goals, we jointly optimize the initialized fingerprint queries to preserve distinctive ownership evidence under ensemble averaging. Specifically, REVEAL reinforces the protected model's response to each query's target label, helping preserve the resulting ownership signal after ensemble averaging. In addition, REVEAL shapes the responses of independently trained reference models to limit interference from unrelated ensemble members. These complementary objectives aim to distinguish ensembles containing the protected model or one of its derived variants from independent ensembles.

Therefore, we define the overall loss over the fingerprint queries as

$$\mathcal{L}^q = \lambda_o \mathcal{L}_o^q + \lambda_r \mathcal{L}_r^q, \tag{1}$$

where $q \in \{\mathrm{SL}, \mathrm{ML}\}$ denotes the single-label or multi-label classification setting. The terms $\mathcal{L}_o^q$ and $\mathcal{L}_r^q$ are the corresponding protected-model and reference-model losses, respectively, and λ_o and λ_r are weights that balance their contributions.

Case 1: Single-label Classification. We define the protected-model and reference-model objectives for single-label classification as follows.

Case 1A: Reinforcing protected-model responses. For each fingerprint query $\hat{x}_i$ with target $\hat{y}_i$, we aim to increase the probability assigned by P to $\hat{y}_i$. Hence, we define the cross-entropy objective:

$$\mathcal{L}_o^{\mathrm{SL}} = -\frac{1}{N} \sum_{i=1}^{N} \log[p_o(\hat{x}_i)]_{\hat{y}_i}, \tag{2}$$

where $[p_o(\hat{x}_i)]_{\hat{y}_i}$ is the probability assigned by P to the fingerprint target $\hat{y}_i$. Minimizing this objective with respect to the queries, while keeping P and the targets fixed, reinforces the naturally occurring erroneous predictions selected during initialization, thus strengthening the protected model's responses to the fingerprint targets.

Case 1B: Shaping reference-model responses. To mitigate interference from unrelated ensemble members, we optimize the fingerprint queries to elicit near-uniform class-probability distributions from reference models trained independently of the protected model P. These responses limit competing class preferences under ensemble averaging. Let $\mathcal{R} = \{R_1, \ldots, R_B\}$ denote the reference set containing B reference models, and $u = \left[\frac{1}{C}, \ldots, \frac{1}{C}\right]$ the uniform distribution over the C classes. At each optimization iteration t, we randomly sample a subset $\mathcal{R}^t \subseteq \mathcal{R}$ and update the queries to drive the sampled models' predictions toward u. We define

$$\mathcal{L}_r^{\mathrm{SL}} = \frac{1}{N|\mathcal{R}^t|} \sum_{i=1}^{N} \sum_{R_b \in \mathcal{R}^t} \mathrm{KL}\left(p_b(\hat{x}_i) \,\|\, u\right), \tag{3}$$

where $p_b(\hat{x}_i)$ is the class-probability vector produced by reference model R_b for fingerprint query $\hat{x}_i$. Minimizing $\mathcal{L}_r^{\mathrm{SL}}$ suppresses class-specific preferences in the reference-model responses. Under probability averaging, an exactly uniform response contributes equally to all classes and thus introduces no competing class preference. Using reference models as surrogates for unrelated ensemble members, this objective aims to reduce class-specific interference with the protected model's target-specific fingerprint responses.

Case 2: Multi-label Classification: We define the protected-model and reference-model objectives for multi-label classification as follows.

Case 2A: Reinforcing protected-model responses. For multi-label classification, the reinforcement objective acts only on the selected target label $\hat{y}_i$, which P already predicts as present on the initial query. We optimize each fingerprint query to drive the probability assigned to this label toward a predefined value ρ_o, satisfying $\rho_o > \tau$. Thus, we define the soft binary cross-entropy objective

$$\mathcal{L}_o^{\mathrm{ML}} = -\frac{1}{N}\sum_{i=1}^{N}\left[\rho_o \log[p_o(\hat{x}_i)]_{\hat{y}_i} + (1-\rho_o)\log\left(1-[p_o(\hat{x}_i)]_{\hat{y}_i}\right)\right]. \tag{4}$$

Minimizing this objective encourages the selected erroneous positive response at $\hat{y}_i$, establishing a desired probability margin of $\rho_o - \tau$ to make a strong target-specific signal for subsequent ensemble aggregation. The remaining label dimensions are left unconstrained.

Case 2B: Shaping reference-model responses. For multi-label classification, we shape the reference-model responses only on the selected target label $\hat{y}_i$ of each fingerprint query. Specifically, we optimize the query so that each sampled reference model assigns the target label $\hat{y}_i$ a predefined value ρ_r below the label decision threshold τ, i.e., $\rho_r < \tau$. We define

$$\mathcal{L}_r^{\mathrm{ML}} = -\frac{1}{N|\mathcal{R}_t|}\sum_{i=1}^{N}\sum_{R_b\in\mathcal{R}_t}\left[\rho_r \log[p_b(\hat{x}_i)]_{\hat{y}_i} + (1-\rho_r)\log\left(1-[p_b(\hat{x}_i)]_{\hat{y}_i}\right)\right]. \tag{5}$$

Minimizing $\mathcal{L}_r^{\mathrm{ML}}$ drives the sampled reference models' target-label probabilities toward ρ_r. $\rho_r < \tau$ encourages these models to remain individually negative on the target label, while placing ρ_r close to τ limits their suppressive effect on the averaged target-label probability when a protected-model signal is present. The remaining label dimensions are unconstrained by this objective.

Optimization Termination. For both classification settings, we randomly resample the reference subset at each optimization iteration, exposing the fingerprint queries to different combinations of reference models. This strategy aims to reduce dependence on any particular subset. Optimization terminates after a predefined number T of iterations. Then, the model owner obtains the confidential fingerprint set $\mathcal{F} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{N}$, where $\hat{x}_i$ is an optimized query and $\hat{y}_i$ is its fixed target class in the single-label setting or selected target label in the multi-label setting.

3.3 Ownership Verification

Given the generated fingerprint set $\mathcal{F} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{N}$, the verifier submits each fingerprint query $\hat{x}_i$ to the black-box suspect ensemble M_s and checks whether its observable prediction matches the corresponding fingerprint target.

Although fingerprint generation uses task-specific objectives, ownership verification follows a unified matching rule in both classification settings. Let $\tilde{y}_s(\hat{x}_i) \subseteq \mathcal{C}$ denote the label set returned by the suspect ensemble M_s for fingerprint query $\hat{x}_i$. This set contains a single predicted class in the single-label setting and all labels predicted as present in the multi-label setting. For the corresponding target label $\hat{y}_i \in \mathcal{C}$, we define the fingerprint-level matching outcome as $z_i = \mathbb{I}\left[\hat{y}_i \in \tilde{y}_s(\hat{x}_i)\right]$, where $\mathbb{I}[\cdot]$ is the indicator function. Thus, $z_i = 1$ when the ensemble's returned label set contains the target label, indicating agreement with the protected model's target behavior for that query; otherwise, $z_i = 0$.

Ownership score. We aggregate the fingerprint-level matching outcomes into an ownership score: $\mathrm{Score}(M_s) = \frac{1}{N}\sum_{i=1}^{N} z_i$, which measures the fraction of fingerprint queries whose predictions match their designated targets. An ensemble containing P or one of its derived variants is expected to match more targets than an ensemble composed entirely of unrelated models. Therefore, a higher score provides stronger evidence of reuse of the protected model or a derived variant. Given a decision threshold τ_o, we report a positive ownership-verification result if $\mathrm{Score}(M_s) \geq \tau_o$; otherwise, reuse is not verified by the fingerprint test.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on four image datasets spanning both classification settings considered in this paper. We use CIFAR-100 (Krizhevsky et al., 2014) and Oxford-IIIT Pet

(PET) (Parkhi et al., 2012) for single-label classification. For multi-label classification, we utilize PASCAL VOC 2012 (VOC) (Everingham et al., 2010) and COCO-Stuff 10K (COCO) (Caesar et al., 2018), where each image may be associated with multiple semantic categories.

Protected and reference models. For each dataset, we train a protected model P and a set of reference models for the same task. The reference models are independently initialized and trained without access to P. At each iteration of fingerprint optimization, we randomly sample one reference model from this pool to compute the reference-model loss. These models are used exclusively for fingerprint generation and are excluded from all evaluation ensembles.

Testing ensembles. We construct a held-out test set of 120 suspect ensembles, consisting of 50 independent ensembles and 70 pirated ensembles. Each ensemble contains K members, with K randomly selected from $\{2, 3, 4, 5\}$. Each independent ensemble is composed entirely of models trained independently from scratch and unrelated to P. Each pirated ensemble contains exactly one reused member, either P itself or a derived variant, together with $K - 1$ independently trained, unrelated members. Of the 70 pirated ensembles, 20 directly reuse P, and the remaining 50 contain variants produced through five model modification techniques, including pruning (PR), fine-tuning (FT), knowledge distillation (KD), noise injection followed by fine-tuning (NFT), and pruning followed by fine-tuning (PFT). We construct 10 ensembles for each modification type.

All members within an ensemble perform the same task. We compute the ensemble output as the element-wise arithmetic mean of their predictive probability vectors. For single-label classification, we select the class with the highest averaged softmax probabilities. For multi-label classification, we apply the predefined threshold τ to each averaged sigmoid probability to obtain binary label predictions.

To introduce architectural diversity, we select independently trained ensemble members from a pool spanning nine architectures: ResNet-18, ResNet-50, ResNet-101, MobileNetV2, MobileNetV3-Large, EfficientNet-B2, EfficientNet-B4, DenseNet-121, and DenseNet-169. Models sharing an architecture use different random initializations and are trained independently of both the protected and the reference models. All test ensembles are reserved for evaluation and are excluded from fingerprint generation and reference-model construction.

Baselines. We compare REVEAL with four representative model fingerprinting approaches: IPGuard (Cao et al., 2021), AKH (Godinot et al., 2025), IrisFP (Geng et al., 2026), and LiteGuard (Yang et al., 2026b). IPGuard, AKH, and IrisFP are designed for single-label classification and are evaluated on CIFAR-100 and PET. LiteGuard supports both classification settings and is evaluated on all four datasets. Each baseline independently generates fingerprints for the same protected model and performs ownership verification on the same suspect ensembles, following its original fingerprint-generation and verification procedures. We use an identical fingerprint query budget across all methods to ensure a consistent comparison.

Evaluation metric. We use the area under the receiver operating characteristic curve (AUC) as the primary metric for ownership verification. Pirated ensembles containing P or one of its derived variants are treated as positive samples, while independent ensembles serve as negative samples. AUC measures how well the ownership scores distinguish these two groups across decision thresholds, with higher values indicating better discrimination. Unless otherwise stated, we report results averaged over five independent runs.

Implementation details. All methods use the same fingerprint query budget of $N = 100$. By default, REVEAL uses a reference pool of $B = 3$ independently trained models, from which one model is randomly sampled at each fingerprint optimization iteration.

For single-label classification, we initialize fingerprints using samples misclassified by P and retain its erroneous top-1 predictions as the fingerprint targets. For multi-label classification, we select samples for which P predicts at least one label absent from the ground truth as present. For each selected sample, we randomly choose one such false-positive label as the fingerprint target.

We set the protected-model and reference-model loss weights to $\lambda_o = \lambda_r = 1.0$. For multi-label classification, we use a label decision threshold of $\tau = 0.5$ and set the protected-model and reference-model target probabilities to $\rho_o = 0.99$ and $\rho_r = 0.45$, respectively. The ownership-decision threshold is set to $\tau_o = 0.5$. Further implementation details are provided in the Appendix.

Method	Scenario	CIFAR-100	PET	VOC	COCO
IPGuard	Standalone	1.000	1.000	–	–
	Ensemble	0.782 (↓ 0.218)	0.719 (↓ 0.281)	–	–
AKH	Standalone	1.000	1.000	–	–
	Ensemble	0.908 (↓ 0.092)	0.861 (↓ 0.139)	–	–
IrisFP	Standalone	1.000	1.000	–	–
	Ensemble	0.816 (↓ 0.184)	0.896 (↓ 0.104)	–	–
LiteGuard	Standalone	1.000	1.000	1.000	1.000
	Ensemble	0.976 (↓ 0.024)	0.919 (↓ 0.081)	0.918 (↓ 0.082)	0.930 (↓ 0.070)
REVEAL	Standalone	1.000	1.000	1.000	1.000
	Ensemble	1.000 (↓ 0.000)	1.000 (↓ 0.000)	0.994 (↓ 0.006)	1.000 (↓ 0.000)

Table 1: REVEAL's verification performance under standalone-model and ensemble scenarios.

4.2 Effectiveness of the Proposed REVEAL

We evaluate the effectiveness of REVEAL for ownership verification under two deployment scenarios: *Standalone* and *Ensemble*. For each fingerprinting method, we generate fingerprints for the same protected model and evaluate the resulting fingerprint set in both scenarios. In the standalone scenario, the verifier directly queries an individual suspect model. In the ensemble scenario, the verifier queries a suspect ensemble and observes only its final aggregated prediction. Comparing these scenarios allows us to assess how ensemble aggregation affects the ownership evidence captured by different existing fingerprinting methods and whether REVEAL preserves this evidence when individual model responses are obscured by ensemble averaging.

For ease of comparison, parenthesized values report the AUC change relative to the corresponding standalone result, with a downward arrow indicating a decrease under ensemble aggregation. All methods use the same fingerprint query budget, and results are averaged over independent runs.

Discriminative effectiveness. Table 1 reports ownership verification performance across four datasets when the protected model is reused directly, either as a standalone model or as an ensemble member. In the standalone setting, all evaluated methods, including our proposed REVEAL, achieve strong discrimination between the protected model and independently trained models. Under ensemble aggregation, IPGuard, AKH, and IrisFP exhibit AUC reductions on both CIFAR-100 and PET, with the largest decrease reaching 0.281, compared with the standalone setting. LiteGuard is less affected, although its AUC on VOC decreases from 1.0 to 0.918. These results show that ensemble aggregation can weaken the discriminative capability of fingerprints that are effective for standalone models.

In contrast, REVEAL largely preserves its standalone verification performance under ensemble reuse. It maintains an AUC of 1.000 on CIFAR-100, PET, and COCO, and achieves 0.994 on VOC, a decrease of only 0.006 from its standalone result. REVEAL matches the strongest applicable baseline on CIFAR-100, PET, and COCO and improves upon the best baseline on VOC by 0.033 AUC. These results demonstrate that REVEAL maintains strong discrimination between pirated and independent ensembles in the evaluated direct-reuse setting, thereby achieving high uniqueness.

Robustness to model modifications. We evaluate REVEAL against five types of model modifications—FT, PR, KD, NFT, and PFT—under both Standalone and Ensemble settings. In the Standalone setting, each derived model is queried directly as the suspect model, with results reported in Table 2. In the Ensemble setting, the corresponding derived model is incorporated alongside independently trained, unrelated members. We construct 10 pirated ensembles per modification type, with results reported in Table 3.

From Table 2, we observe that REVEAL is competitive in the standalone setting, matching or outperforming the strongest baseline in most model-modification cases. We further find from Table 3 that under ensemble reuse, REVEAL achieves AUCs ranging from 0.936 to 1.000 across the 16 dataset-modification combinations involving FT, PR, NFT, and PFT, matching or outperforming the strongest baseline in 15 of these cases. For example, on VOC, REVEAL achieves AUCs of 0.989 and 0.994 under PR and PFT, respectively, compared with 0.77 and 0.876 for LiteGuard. These results demonstrate REVEAL's robustness to these four modifications under ensemble reuse. KD

remains more challenging on some datasets, consistent with the limitations observed for existing fingerprinting methods.

Dataset	Method	FT	PR	KD	NFT	PFT
CIFAR-100	IPGuard	0.404	1.000	0.765	0.419	0.421
	AKH	0.771	1.000	0.859	0.774	0.781
	IrisFP	**1.000**	0.965	0.490	**1.000**	**1.000**
	LiteGuard	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	REVEAL	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
PET	IPGuard	0.976	0.984	0.591	0.956	0.988
	AKH	0.986	0.981	0.642	0.978	0.968
	IrisFP	0.982	0.991	0.507	0.986	0.984
	LiteGuard	0.968	**1.000**	0.573	0.981	**1.000**
	REVEAL	**0.991**	**1.000**	**0.714**	**1.000**	**1.000**
VOC	LiteGuard	0.976	0.967	**0.821**	0.966	0.973
	REVEAL	**1.000**	**1.000**	0.732	**1.000**	**1.000**
COCO	LiteGuard	0.968	0.951	**0.867**	0.985	0.970
	REVEAL	**1.000**	**1.000**	0.701	**1.000**	**1.000**

Table 2: The verification performance under different model modifications in the standalone setting.

Dataset	Method	FT	PR	KD	NFT	PFT
CIFAR-100	IPGuard	0.425	0.722	0.717	0.438	0.396
	AKH	0.683	0.956	0.769	0.583	0.694
	IrisFP	0.991	0.531	0.461	0.957	0.914
	LiteGuard	0.966	0.991	0.785	0.921	0.942
	REVEAL	**1.000**	**1.000**	**0.994**	**1.000**	**1.000**
PET	IPGuard	0.400	0.469	0.436	0.378	0.214
	AKH	0.750	0.636	0.275	0.756	0.806
	IrisFP	0.747	0.650	**0.672**	0.753	0.672
	LiteGuard	0.874	0.913	0.351	0.901	0.908
	REVEAL	**0.936**	**0.994**	0.464	**0.992**	**1.000**
VOC	LiteGuard	0.891	0.770	**0.343**	0.906	0.876
	REVEAL	**0.986**	**0.989**	0.231	**0.994**	**0.994**
COCO	LiteGuard	0.889	0.923	0.438	0.883	0.880
	REVEAL	**0.936**	**1.000**	**0.465**	**0.994**	**0.992**

Table 3: The verification performance under different model modifications in the ensemble setting.

5 CONCLUSIONS

In this paper, we investigate an underexplored challenge in model ownership verification: unauthorized reuse of a protected DNN or one of its derived variants within a black-box ensemble. To address this issue, we propose REVEAL, a fingerprinting framework for black-box ensembles supporting both single-label and multi-label classification. Specifically, REVEAL first initializes fingerprint queries from samples on which the protected model naturally makes prediction errors, retaining one erroneous predicted label per sample as its fixed target. Then, it further optimizes these queries to jointly reinforce the protected model's target-specific behavior and shape the responses of independently trained reference models. This design helps preserve ownership evidence under ensemble averaging while limiting interference from unrelated members. Ownership verification follows a unified rule that checks whether each returned label set contains the corresponding fingerprint targets. Extensive experiments across four datasets and common model modifications show that REVEAL mitigates verification degradation caused by ensemble aggregation and effectively detects reuse of both protected models and their derived variants. However, the results also reveal limitations in detecting reuse of derivatives obtained through knowledge distillation, highlighting a remaining challenge for ensemble-based ownership verification.

AI USE STATEMENT

In this paper, we have not used generative AI tools for any tasks with required disclosure. We used generative AI tools only for editing and improving the paper readability. We reviewed all AI-assisted edits to ensure that they preserved the paper's technical meaning and scientific content. We take full responsibility for the final paper, including all AI-assisted text.

ETHICS STATEMENT

This work studies model ownership verification to support the detection of unauthorized model reuse in black-box ensembles. All experiments use publicly available benchmark datasets and models trained or evaluated in a locally controlled environment. The study does not involve human-subject recruitment or the collection of private user data, and no experiments target deployed systems or third-party services.

A potential dual-use concern is that evaluating fingerprint robustness under model modifications may help adversaries understand how to evade ownership verification. Our evaluation applies standard modification techniques, including fine-tuning, pruning, and knowledge distillation, to controlled experimental models to assess REVEAL's verification performance. These procedures serve legitimate purposes in model development; their use alone does not imply unauthorized activity.

REVEAL provides behavioral evidence that an ensemble contains a protected model or one of its derived variants. Verification results remain subject to false positives and false negatives and do not independently establish whether reuse was authorized. Therefore, conclusions about unauthorized reuse should consider fingerprint evidence alongside authorization records and other relevant information.

REFERENCES

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE/CVF international conference on computer vision (CVPR)*, pp. 1209–1218, 2018.

Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pp. 14–25, 2021.

Yufei Chen, Chao Shen, Cong Wang, and Yang Zhang. Teacher model fingerprinting attacks against transfer learning. In *Proceedings of 31st USENIX Security Symposium*, pp. 3593–3610, 2022.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.

Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88: 303–338, June 2010.

Mudasir A Ganaie, Minghui Hu, Ashwani Kumar Malik, Muhammad Tanveer, and Ponnuthurai N Suganthan. Ensemble deep learning: A review. *Engineering applications of artificial intelligence*, 115:105151, 2022.

Ziye Geng, Guang Yang, Yihang Chen, and Changqing Luo. Irisfp: Adversarial-example-based model fingerprinting with enhanced uniqueness and robustness. In *Proceedings of the IEEE/CVF international conference on computer vision (CVPR)*, 2026.

Augustin Godinot, Erwan Le Merrer, Camilla Penzo, François Taïani, and Gilles Trédan. Queries, representation & detection: The next 100 model fingerprinting schemes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16817–16825, 2025.

Jiyang Guan, Jian Liang, and Ran He. Are you stealing my model? sample correlation for fingerprinting deep neural networks. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 3657–3658, 2022.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get m for free. In *International Conference on Learning Representations*, 2017.

Alex Krizhevsky, Vinod Nair, Geoffrey Hinton, et al. The CIFAR-100 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5):2, 2014.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Weixing Liu and Shenghua Zhong. MarginFinger: Controlling generated fingerprint distance to classification boundary using conditional gans. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, pp. 129–136, 2024.

Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. In *2021 International Conference on Learning Representations*, 2021.

Thibault Maho, Teddy Furon, and Erwan Le Merrer. Fingerprinting classifiers with benign inputs. *IEEE Transactions on Information Forensics and Security*, 18:5459–5472, 2023.

Xudong Pan, Yifan Yan, Mi Zhang, and Min Yang. MetaV: A meta-verifier approach to task-agnostic model fingerprinting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1327 – 1336, 2022.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13420–13429, 2022.

Bolin Shen, Zhan Cheng, Neil Zhenqiang Gong, Fan Yao, and Yushun Dong. CREDIT: Certified ownership verification of deep neural networks against model extraction attacks. In *Forty-third International Conference on Machine Learning*, 2026.

Tianlong Xu, Chen Wang, Gaoyang Liu, Yang Yang, Kai Peng, and Wei Liu. United we stand, divided we fall: Fingerprinting deep neural networks via adversarial trajectories. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pp. 6929–6932, 2024.

Tianlong Xu, Zixiong Wang, Lishuai Hou, Gaoyang Liu, Chen Wang, and Xiaoyi Fan. Fingerprinting pre-trained encoders under arbitrary downstream fine-tuning via adversarial shifting. In *Forty-third International Conference on Machine Learning*, 2026a.

Tianlong Xu, Zixiong Wang, Gaoyang Liu, Jian Chen, Ahmed M. Abdelmoniem, and Chen Wang. Breaking the boundary barrier: Robust model fingerprinting via unlearnable examples in model-parameter space. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1739–1750, 2026b.

Guang Yang, Ziye Geng, Yihang Chen, and Changqing Luo. Fingerprinting deep neural networks for ownership protection: An analytical approach. *arXiv preprint arXiv:2603.21411*, 2026a.

Guang Yang, Ziye Geng, Yihang Chen, and Changqing Luo. Liteguard: Efficient task-agnostic model fingerprinting with enhanced generalization. *arXiv preprint arXiv:2603.24982*, 2026b.

Kan Yang, Run Wang, and Lina Wang. MetaFinger: Fingerprinting the deep neural networks with meta-training. In *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 776–782, 2022.

Zhuomeng Zhang, Fangqi Li, Hanyi Wang, and Shi-Lin Wang. Boosting the uniqueness of neural networks fingerprints with informative triggers. In *Proceedings of The 39th Annual Conference on Neural Information Processing Systems*, pp. 85072–85095, 2025.

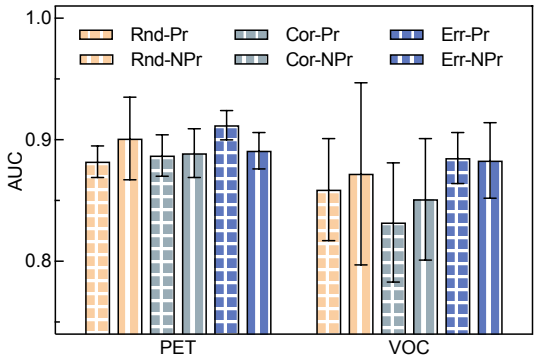

Figure 3: Impact of fingerprint-query initialization and target selection on REVEAL.

A Appendix

A.1 Ablation Studies

A.1.1 The impact of fingerprint initialization and target selection.

We investigate how fingerprint initialization and target selection affect REVEAL's verification performance. We consider three initialization strategies: Err, which uses the natural-error samples defined in Sec. 3.2.1, Cor, which utilizes correctly predicted samples, and Rnd, which randomly selects samples regardless of prediction correctness.

For target selection, we consider Pr and NPr. In the single-label setting, Pr uses the class predicted by the protected model, while NPr randomly selects a different class from the predicted class. In the multi-label setting, Pr randomly selects a label predicted as present, whereas NPr randomly selects a label predicted as absent. For Err-Pr, target selection is further restricted to the natural errors defined in Sec. 3.2.1: the erroneous top-1 prediction in the single-label setting or a false-positive label in the multi-label setting. Err-Pr is the default configuration of REVEAL.

Combining the three initialization strategies with the two target-selection strategies yields six configurations: Rnd-Pr, Rnd-NPr, Cor-Pr, Cor-NPr, Err-Pr, and Err-NPr. All other components and hyperparameters remain unchanged. We evaluate these configurations on PET and VOC, representing the single-label and multi-label classification, respectively, and report AUC under ensemble aggregation.

As shown in Fig. 3, Err-Pr achieves the highest mean AUC on both PET and VOC, reaching 0.912 and 0.885, respectively. With Pr retained, replacing Err initialization with Cor or Rnd reduces AUC on both datasets. Replacing Pr with NPr under Err initialization also provides no improvement. These results support the default Err-Pr configuration, which combines natural-error initialization with targets drawn from the protected model's erroneous predictions.

B Hyperparameters Analysis

B.1 The impact of the reference set size B

We evaluate the effect of the reference set size B, i.e., the number of independently trained reference models used for fingerprint generation, by varying $B \in \{1, 2, 3, 4\}$. At each optimization iteration, one reference model is randomly sampled from this set and uses its responses to compute the response-shaping loss. As shown in Fig. 4, increasing B generally improves verification performance. On PET, the AUC increases from 0.891 at $B = 1$ to 0.906 at $B = 2$ and 0.916 at $B = 4$. Similarly, on VOC, the AUC improves from 0.872 to 0.885 and 0.891, respectively. Moving from one to two reference models provides most of the improvement observed at $B = 4$, with smaller gains from further expansion. These results suggest that exposing fingerprint optimization to multiple independently trained models improves generalization to unseen ensemble members, while a small reference set captures much of the observed benefit.

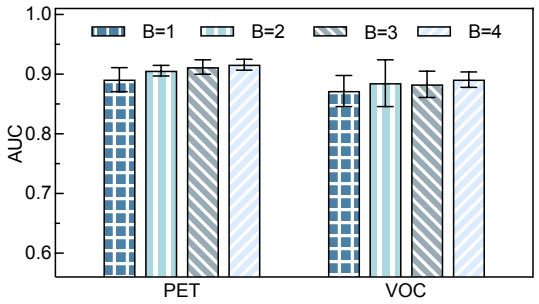

Figure 4: Impact of the reference set size B on REVEAL's performance on PET and VOC.

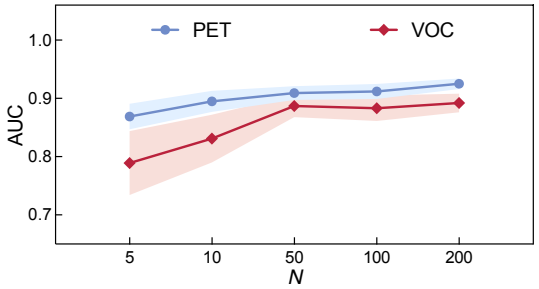

Figure 5: Impact of the number of fingerprints N on REVEAL's performance on PET and VOC.

B.2 Impact of the number of fingerprints N

We assess the impact of the number of fingerprints N on REVEAL's verification performance by varying $N \in \{5, 10, 50, 100, 200\}$. As shown in Fig. 5, increasing N generally improves the AUC on both PET and VOC. On PET, the AUC increases from 0.869 at $N = 5$ to 0.909 at $N = 50$ and 0.925 at $N = 200$. Similarly, on VOC, the AUC rises from 0.789 at $N = 5$ to 0.887 at $N = 50$, after which it exhibits only minor fluctuations, reaching 0.892 at $N = 200$. These results suggest that larger fingerprint sets improve verification performance, while a moderate set of 50 fingerprints captures most of the observed gain. Further increasing the set size provides smaller additional benefits, particularly on VOC.

B.3 The impact of coefficients λ_o and λ_r

We evaluate the effect of the two loss weights in Eq. 1, where λ_o controls the protected-model response reinforcement and λ_r controls the reference-model response shaping. Fig. 6 reports the results for different weight configurations, with a/b denoting $\lambda_o = a$ and $\lambda_r = b$. Under the $1.0/0.2$, $1.0/1.0$, and $0.2/1.0$ configurations, the AUCs on PET are 0.907, 0.912, and 0.899, respectively, while those on VOC are 0.868, 0.883, and 0.863. These modest variations suggest that REVEAL is relatively insensitive to the weighting of the two objectives within the evaluated range when both remain active. However, omitting either objective degrades verification performance, with a substantially larger decline when protected-model response reinforcement is removed. These findings support retaining both objectives while allowing flexibility in their relative weights.

B.4 The impact of the target probability ρ_o and ρ_r

We investigate the impact of the protected-model target probability ρ_o in the multi-label setting by varying $\rho_o \in \{0.65, 0.75, 0.85, 0.95, 0.99\}$. As shown in Fig. 7, increasing ρ_o consistently improves verification performance on both VOC and COCO. On VOC, the AUC increases from 0.613 at $\rho_o = 0.65$ to 0.874 at $\rho_o = 0.95$ and 0.883 at $\rho_o = 0.99$. Similarly, on COCO, the AUC improves from 0.492 to 0.886 and 0.902, respectively. The gains become relatively small when ρ_o increases from 0.95 to 0.99, suggesting diminishing improvements at the upper end of the tested range. These

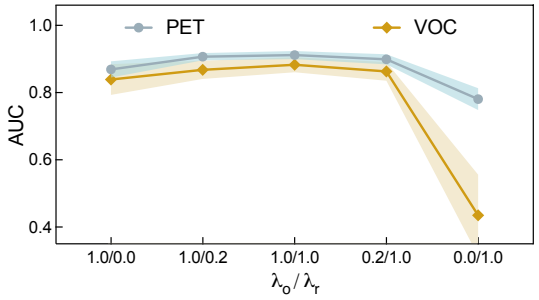

Figure 6: Impact of loss weights λ_o and λ_r on REVEAL's performance on PET and VOC.

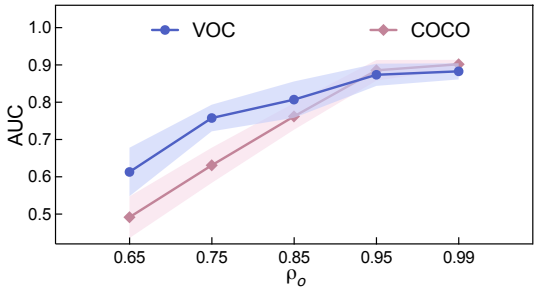

Figure 7: Impact of the protected-model target probability ρ_o on REVEAL's performance on VOC and COCO.

results indicate that a higher ρ_o strengthens the protected model's target-label response with a larger margin above the decision threshold, helping preserve the target behavior after ensemble averaging.

We investigate the impact of the reference-model target probability ρ_r in the multi-label setting by varying $\rho_r \in \{0.0, 0.25, 0.45, 0.75\}$. As shown in Fig. 8, increasing ρ_r from 0.0 to 0.45 consistently improves verification performance on both VOC and COCO. On VOC, the AUC increases from 0.824 at $\rho_r = 0.0$ to 0.883 at $\rho_r = 0.45$, while on COCO it increases from 0.858 to 0.902. In contrast, further increasing ρ_r to 0.75 reduces the AUC to 0.870 and 0.865, respectively. These results indicate that setting ρ_r near the decision threshold while remaining below the decision threshold provides the most favorable reference-model response. An excessively small ρ_r can overly suppress the target-label probability during ensemble aggregation, whereas a value above the threshold may cause unrelated models to independently predict the target label as present. These results suggest that setting ρ_r slightly below the decision threshold provides a favorable trade-off for reference-model response shaping.

B.5 The impact of the number of ensemble members K

We evaluate the impact of the number of ensemble members K by varying $K \in \{2, 4, 6, 8, 10\}$. As shown in Fig. 9, the verification AUC gradually decreases on both PET and VOC as the ensemble size K increases. On PET, the AUC decreases from 0.936 at $K = 2$ to 0.888 at $K = 6$ and 0.858 at $K = 10$. Similarly, on VOC, the AUC decreases from 0.939 to 0.861 and 0.835, respectively. This trend is consistent with the increasing dilution of the reused model's contribution under equal-weight averaging as more independently trained, unrelated members are added in an ensemble. Nevertheless, REVEAL maintains discriminative power at the largest tested ensemble size of $K = 10$, showing that its fingerprints remain informative despite the increased dilution of ownership evidence.

C Additional Implementation Details

Hyperparameters. We optimize the fingerprint queries using Adam with a learning rate of 0.01 for $T = 200$ iterations.

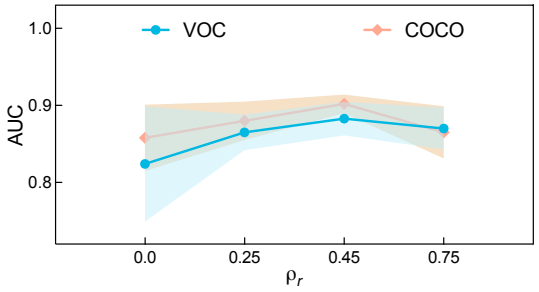

Figure 8: The impact of the reference-model target probability ρ_r on REVEAL's performance on VOC and COCO.

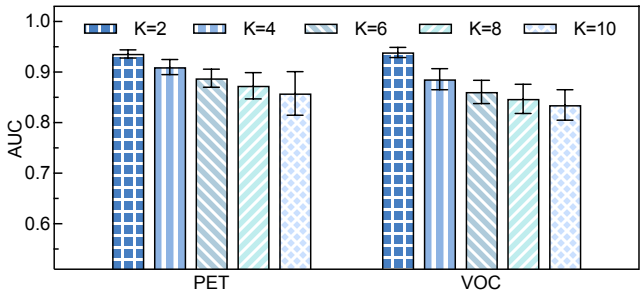

Figure 9: Impact of the number of ensemble members K on REVEAL's performance on PET and VOC.

Protected model and reference model set. A ResNet-18 model is trained as the protected model P for each dataset. The reference set is formed by additional ResNet-18 models independently trained from different random initializations on the same dataset, without access to the parameters, checkpoints, or outputs of P. These reference models are used only during fingerprint generation and are never included as members of the evaluation ensembles. Likewise, all independently trained models used to construct the evaluation ensembles are separately trained and remain independent of both P and the reference set. The only intentional dependence on P occurs in pirated ensembles, where P itself or one of its derived variants is included as the reused member.

Evaluation ensemble sets. We construct two held-out sets of suspect ensembles for evaluation: 50 independent ensembles and 70 pirated ensembles. Each ensemble contains $K \in \{2, 3, 4, 5\}$ members, with K randomly selected for each ensemble. To ensure architectural diversity, the independently trained members are drawn from nine architectures: ResNet-18, ResNet-50, ResNet-101, MobileNetV2, MobileNetV3-Large, EfficientNet-B2, EfficientNet-B4, DenseNet-121, and DenseNet-169. Each such model is trained independently from scratch on the corresponding dataset, and models sharing the same architecture use different random initializations.

The two evaluation sets differ only in whether they contain a model related to the protected model P. An independent ensemble consists entirely of independently trained members. A pirated ensemble contains exactly one reused member, corresponding either to P itself or to a derived variant P', while all remaining $K - 1$ members are independently trained and unrelated to P. Among the 70 pirated ensembles, 20 directly include P, while the other 50 include variants obtained through five model modifications, with 10 ensembles constructed for each modification type.

Model modifications. We consider five representative model modification techniques to derive variants of P: fine-tuning (FT), pruning (PR), knowledge distillation (KD), noise injection followed by fine-tuning (NFT), and pruning followed by fine-tuning (PFT). FT further trains P on its original task without changing its architecture. PR applies global unstructured pruning at sparsity levels ranging from 10% to 90% in increments of 10%. KD trains student models with architectures either identical to or different from that of P to mimic its output behavior, using a distillation temperature of 1. NFT first perturbs model parameters using Gaussian noise scaled by the standard deviation

of each parameter tensor and then fine-tunes the perturbed model. PFT first applies 70% global unstructured pruning and then fine-tunes the resulting model to recover predictive performance.

D THE COMPUTING INFRASTRUCTURE

All experiments were run on a server equipped with a 40 GB NVIDIA A100 GPU and a 3.30 GHz Intel Xeon Gold 6246 CPU. REVEAL was implemented using Python 3.9.21, PyTorch 2.5.1, and CUDA 12.1.

E RELATED WORK

Model fingerprinting has emerged as a promising, non-intrusive approach to verifying model ownership without modifying the protected model (Cao et al., 2021; Yang et al., 2022; Chen et al., 2022; Shen et al., 2026). Fingerprints are designed to capture distinctive behavioral characteristics that are consistent across derived variants while distinguishing them from independently trained models. So far, many fingerprinting approaches have been developed. These approaches differ primarily in how fingerprint queries are constructed and how model responses are interpreted for verification.

Several works construct fingerprints by selecting informative samples that naturally expose model-specific behaviors without explicitly optimizing the fingerprint queries. For example, Maho et al. (2023) select discriminative benign inputs for ownership verification. Guan et al. (2022) propose sample correlation-based fingerprinting, which compares correlations among model outputs on misclassified or CutMix-augmented samples. Godinot et al. (2025) present the Anna Karenina Heuristic (AKH) alongside their QuRD framework. This method utilizes naturally misclassified samples as queries and compares suspect-model responses with the protected model's erroneous predictions. These approaches select fingerprints from existing data distribution without further optimizing them for ownership separation, consequently limiting their discriminative power.

Some other works actively optimize fingerprint queries to expose distinctive decision-boundary behavior. Cao et al. (2021) construct fingerprints around the protected model's decision boundaries and verify ownership through prediction consistency. Peng et al. (2022) use universal adversarial perturbations to characterize global properties of the protected model's decision boundaries across multiple inputs. Xu et al. (2024) constructs adversarial trajectories consisting of progressively adversarial samples and use model responses along these trajectories to improve robustness to changes in decision boundaries.

Very recent works further investigate fingerprint uniqueness, robustness to model modifications, and verification generalization. Xu et al. (2026b) improves fingerprint robustness to model modifications by constructing unlearnable examples whose low-loss behavior remains stable across parameter variations. Geng et al. (2026) construct composite fingerprints by jointly leveraging information from the intersection of multiple decision boundaries and the collective behavior of composite samples, leading to better separation between pirated and independently trained models. Yang et al. (2026a) theoretically studies how a fingerprint's location relative to the decision boundary affects its uniqueness and robustness, and uses this analysis to guide fingerprint optimization. Xu et al. (2026a) improves verification generalization across downstream classification tasks by constructing adversarially shifted fingerprints in the encoder feature space, making verification independent of task-specific label spaces. Yang et al. (2026b) improves verification generalization with limited auxiliary models through checkpoint-based model augmentation and fingerprint-specific local verifiers.

These works primarily consider standalone verification, where the suspect model's response to each fingerprint query is directly observable. This setting differs from verifying whether a protected model or one of its derived variants is incorporated into an ensemble. In the latter setting, the verifier observes only the ensemble's aggregated prediction, while the reused model's individual response remains hidden. Predictions from other ensemble members can attenuate or alter the ownership-specific behavior elicited by a fingerprint. Consequently, fingerprints developed for standalone deployment may become less discriminative after ensemble aggregation.

The task's prediction mechanism introduces an additional challenge. Many classification-oriented fingerprinting methods focus on single-label tasks. Although LiteGuard (Yang et al., 2026b) supports both single-label and multi-label classification, it targets standalone suspect models. For en-

semble verification, these two task settings impose different requirements for preserving ownership evidence: single-label classification selects the class with the highest aggregated probability, whereas multi-label classification independently thresholds the aggregated probability of each label. Hence, effective fingerprint design must account for both ensemble aggregation and the underlying decision rule to preserve ownership-indicative behavior in the observable output.