

TP-CRIV: A Framework for Third-Party Challenge-Response Identity Verification of AI Models

Teruki Sano, Minoru Kuribayashi, Masao Sakai, Shuji Isobe, Eisuke Koizumi, Zhang Zhang, and Satoru Matsumoto

Abstract—Artificial intelligence (AI) models are increasingly deployed through remote services, making model misappropriation a growing concern. Existing approaches, including watermarking, fingerprinting, and model similarity analysis, primarily rely on predefined evidence or direct behavioral comparison and do not explicitly evaluate whether the claimant currently possesses and can utilize model-dependent information relevant to the claimed model identity.

In this paper, we propose Third-Party Challenge-Response Identity Verification (TP-CRIV) for AI models. TP-CRIV targets a third-party verification setting in which the verifier has neither white-box nor API access to the claimant’s model, can interact with the suspicious deployed service only through its ordinary black-box inference interface, and does not require protocol-specific cooperation from the service provider. Under these constraints, the framework enables the verifier to obtain empirical evidence as to whether the claimant locally possesses a model satisfying a predeclared identity relative to the deployed model. Verification is conducted under fresh, previously undisclosed requirements and network isolation, so that the demonstrated capability cannot rely on online external assistance after challenge disclosure. The resulting evidence is interpreted relative to independently specified and calibrated matching and non-matching operating situations and is statistical rather than cryptographic. We instantiate TP-CRIV for CNN image classifiers using probability-control-based witness generation. Experiments on ten ImageNet-pretrained TorchVision models demonstrate clear same/cross-model separation and finite-challenge verification using independently calibrated thresholds.

Index Terms—AI security, Challenge-response, Model identity verification, Model-dependent capability verification, Machine learning as a service (MLaaS).

I. INTRODUCTION

Artificial intelligence (AI) models support a wide range of applications, while their development can require substantial computational resources, expert knowledge, and financial investment. Protecting the intellectual property (IP) of AI

models has therefore become an important issue [1], [2], particularly as model theft and unauthorized reuse remain practical concerns [3], [4].

At the same time, many models are deployed through Machine Learning as a Service (MLaaS), where users interact with remote APIs rather than obtaining the models themselves. In such settings, the parameters, architectures, and training records of a suspicious deployed model are generally unavailable to external parties. Consequently, even if a claimant identifies a suspicious MLaaS, an independent third party cannot directly compare the claimant’s model with the deployed model to assess their relationship. Moreover, without sufficiently credible preliminary evidence supporting the claimant’s allegation, the third party may have difficulty justifying a request for protocol-specific cooperation, model disclosure, or other additional assistance from the MLaaS provider. Under such circumstances, the claimant must first provide convincing technical evidence that supports the suspicion of model misappropriation and justifies further investigation. Therefore, the fundamental research question considered in this work is **how an independent third party can obtain credible preliminary evidence of possible model misappropriation without white-box access to either model and without requiring protocol-specific cooperation from the suspicious deployed service**. Rather than attempting to measure global similarity between two inaccessible models, we ask whether the third party can issue fresh requirements whose solutions depend strongly on model-specific behavior of the suspicious service and evaluate whether the claimant can satisfy those requirements using a locally available model.

From the perspective of this research question, existing model-protection approaches mainly use predefined model-specific evidence, as in watermarking and fingerprinting [1], [5]–[20], or compare model behavior with a reference model [21]–[24]. Proof-based verifiable-inference approaches address a different relation: rather than comparing two models, they verify that a reported computation was correctly performed using a specified or committed model [25]–[30]. These approaches provide valuable ownership, identity, similarity, or computation-integrity evidence, but they address a different question from whether a claimant can use a locally available model to satisfy fresh third-party requirements concerning a suspicious deployed service.

To address this problem, we propose Third-Party Challenge-Response Identity Verification (TP-CRIV) for AI models. TP-

Teruki Sano is with the Graduate School of Information Sciences, Tohoku University, Japan.

Minoru Kuribayashi, Masao Sakai, Shuji Isobe, Eisuke Koizumi, Zhang Zhang, and Satoru Matsumoto are with the Center for Data-driven Science and Artificial Intelligence, Tohoku University, Japan.

Corresponding author: Teruki Sano (e-mail: sano.teruki.r2@dc.tohoku.ac.jp).

This study was supported by the JSPS KAKENHI (25K15225, 26H02540) and Support Center for Advanced Telecommunications Technology Research (SCAT), Japan.

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

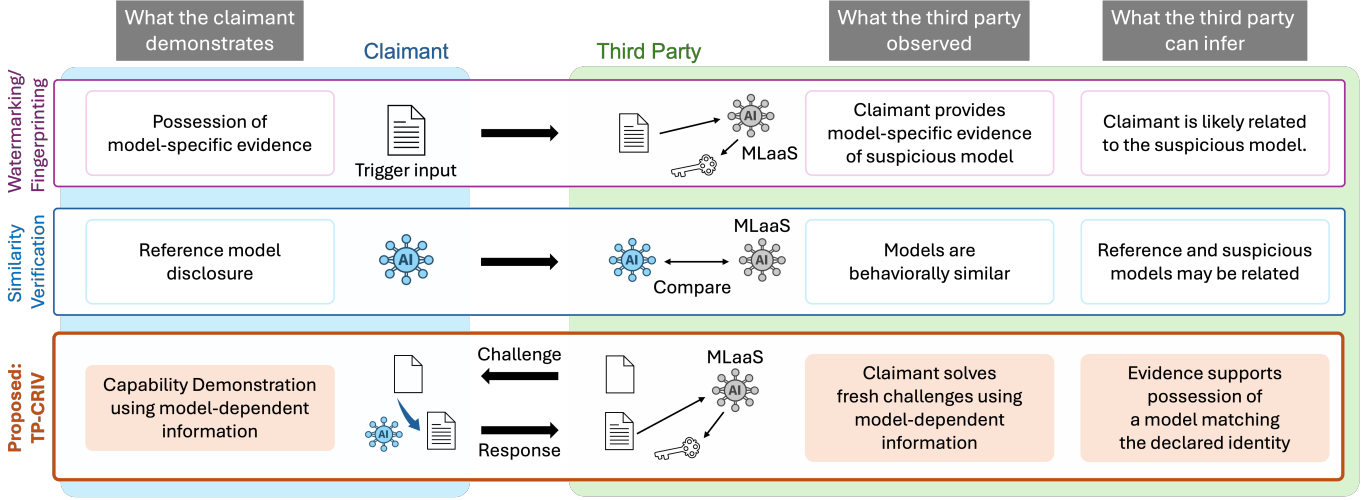


Fig. 1. Comparison of model-to-model verification approaches from the perspective of the evidence available to an independent third party and the resulting inference about the relationship between a claimant’s model and a suspicious deployed model.

CRIV enables an independent third party to obtain empirical evidence as to whether a claimant possesses a model satisfying a specified identity relative to a suspicious deployed model, while requiring neither white-box nor API access to the claimant’s model and only ordinary black-box access to the suspicious MLaaS. Fig. 1 compares this verification objective with those of existing model-verification approaches. Specifically, for each intended verification problem, the verifier first specifies two disjoint operating situations representing matching and non-matching configurations. These situations define the objective-dependent model identity independently of the challenge-response score or acceptance outcome. The verifier then selects a model-dependent property, issues fresh property-demanding challenges, and evaluates the returned witnesses through the deployed service. An operational threshold is calibrated using independent reference configurations from the declared matching and non-matching situations. Freshness and network isolation exclude online external assistance after challenge disclosure. Within the model-based response-generation scope, matching-consistent challenge-solving performance provides an empirical basis for inferring that the claimant possesses a model satisfying the specified identity. This inference is statistical rather than cryptographic and does not establish training provenance or legal ownership.

To clarify the role of TP-CRIV in practice, we view model-misappropriation verification as a two-stage process.

Step 1: Preliminary Evidence Assessment. Suppose that a claimant asserts that it possesses a model identical to the model deployed by a suspicious MLaaS. At this stage, however, an independent third-party verifier may have insufficient evidence to determine whether the claim is credible. Under such circumstances, it may also be difficult to justify requiring the suspicious MLaaS provider to participate in a dedicated verification procedure. The verifier must therefore first assess the technical credibility of the claimant’s misappropriation claim using externally observable evidence and determine whether the case warrants a more rigorous subsequent investigation.

Step 2: Formal Ownership Investigation. If sufficient suspicion is established, a formal investigation may employ stronger verification conditions, including direct access to the relevant models or protocol-specific cooperation from the involved parties, together with ownership-related evidence such as development records, timestamps, and ownership documents. Based on this technical and ownership-related evidence, the verifier determines whether model misappropriation or ownership infringement has occurred.

TP-CRIV addresses the first stage by providing preliminary technical evidence about possession of a model satisfying the declared identity.

A. Our Contributions

The main contributions of this paper are summarized as follows:

- We propose TP-CRIV for preliminary third-party verification in suspicious MLaaS settings where the verifier has neither white-box nor API access to the claimant’s model and the deployed service provides only its ordinary black-box inference interface. Under these conditions, TP-CRIV enables the verifier to obtain empirical evidence as to whether the claimant possesses a model satisfying the objective-dependent identity relative to the deployed model, without requiring protocol-specific cooperation from the service provider.
- To demonstrate the practical feasibility of TP-CRIV, we instantiate the framework for CNN-based image classification using local input–probability geometry and probability-control-based witness generation [31]. Experiments with ten ImageNet-pretrained TorchVision CNN models demonstrate clear separation between the declared matching and non-matching configurations and show that calibration on separate models enables finite-challenge decisions on held-out models.

B. Organization

The remainder of this paper is organized as follows. Section II reviews related studies on model protection and verification, including ownership, identity, similarity, and proof-based verifiable inference. Section III presents the TP-CRIV framework and its verification principle. Section IV presents a CNN-based instantiation and feasibility evaluation. Section V concludes the paper and outlines future work.

II. RELATED WORK

This section reviews existing approaches relevant to model protection and verification, focusing on the evidence available to an independent third party and the conclusions that can be drawn from that evidence.

A. Model-Specific Evidence for Verification

Model watermarking and fingerprinting support ownership or identity claims using model-specific information. Although they construct such information in different ways, verification ultimately evaluates evidence associated with a particular model.

1) *Model Watermarking*: Model watermarking embeds proprietary information into an AI model and verifies ownership by detecting or extracting the embedded watermark [1], [5]–[12]. Representative approaches include trigger-based watermarks that induce predefined behaviors for specific inputs [7], [9], [11], [12] and methods that recover embedded messages from model outputs [10], [12]. In black-box or gray-box settings, such evidence can be verified through observable model responses without exposing the model itself. Successful verification therefore demonstrates that the presented watermark-related evidence is consistent with the suspicious model. Under the assumption that this evidence remains exclusive to the legitimate owner, such consistency can support an ownership claim. However, if the watermark information is disclosed or acquired by another party, the exclusivity assumption may be weakened. Moreover, because the watermark-related information is generally defined before verification, the verification primarily evaluates predefined evidence rather than whether the claimant currently possesses a model capable of producing fresh model-dependent responses.

2) *Model Fingerprinting*: Model fingerprinting identifies models through intrinsic model-dependent behavior without modifying the target model [13]–[20]. Representative approaches use decision-boundary characteristics [13]–[16], model-specific test examples [17], universal adversarial perturbations [18], or model-sensitive behavior of large language models [19]. Such fingerprints provide evidence for determining whether the behavior of a suspicious model is consistent with model-specific information derived from a reference model. However, fingerprint evidence can generally be constructed before the verification session and may subsequently be presented or reused without requiring the claimant to demonstrate access to the model from which it was derived. Consequently, successful fingerprint verification does not by itself establish that the claimant currently possesses a model satisfying the claimed identity.

a) *Limitation of Model-Specific Evidence*: Watermarking and fingerprinting provide valuable ownership and identity evidence, but both primarily verify consistency with model-specific evidence prepared before verification. Accordingly, a third party can evaluate the relationship between the presented evidence and the suspicious model, but cannot directly determine from that evidence alone whether the claimant currently possesses a model capable of generating corresponding model-dependent behavior. Furthermore, the evidence itself may be proprietary or valuable and may not always be desirable to disclose to an external verifier. These limitations motivate verification that evaluates fresh model-dependent capability rather than only possession or disclosure of predefined evidence.

B. Model Similarity Verification

Several studies verify model similarity or provenance by comparing observable or internal model behavior [21]–[24]. Representative approaches include test-case-based comparison [21], [22], model lineage analysis using internal information [23], and provenance testing based on observable outputs [24]. These methods can provide strong evidence about the relationship between models, but generally assume that the models to be compared are available to the verifier at least through their observable behavior, and some methods use white-box reference-model information to construct model-sensitive test cases [22]. In practical settings, however, the claimant may also be unable to provide its model to a third party, for example because the model is proprietary or too large to make such access practical. Under such conditions, direct model-similarity verification becomes difficult.

TP-CRIV targets a different operational setting in which the claimant's model is not made available to the verifier, even through query access, and the suspicious deployed service is accessible only through its ordinary black-box interface.

C. Proof-based Verifiable Inference

Proof-based verifiable-inference methods address a different verification objective from model-to-model identity verification. Their primary goal is to prove that a reported output was correctly computed from a given input using a specified or committed model, without necessarily revealing the model parameters [25]–[30]. Thus, these methods primarily establish a cryptographic relation between a model and its computation, rather than directly evaluating the relationship between a claimant-held model and a separately deployed suspicious model.

In typical verifiable-inference settings, the model-holding service acts as the prover and participates in a dedicated verification procedure, for example by committing to the model and generating an inference proof. Such mechanisms can provide strong guarantees when this participation is available. However, as discussed above, they become difficult to apply when requiring the suspicious MLaaS provider to participate in a dedicated verification procedure is impractical. TP-CRIV instead evaluates the relationship between two separately held models through only ordinary black-box access to the suspicious MLaaS, enabling the verifier to infer the claimant's

TABLE I
COMPARISON OF VERIFICATION OBJECTIVES, EVIDENCE, AND OPERATIONAL REQUIREMENTS.

Method	Model-Specific Evidence	Similarity Verification	Proof-Based Verification	Proposed: TP-CRIV
Primary objective	Support ownership or model-identity claims	Estimate or verify similarity between models	Verify correctness of a specified or committed computation	Third-party verification of objective-dependent identity
Directly evaluated evidence	Consistency with predefined watermark or fingerprint evidence	Behavioral or internal consistency between compared models	Proof that a defined computation or relation is satisfied	Responses to fresh verifier-issued requirements
Typical approach	Watermark reproduction, extraction, or fingerprint response evaluation	Reference-model-based comparison	Proof generation by the model-holding party	Challenge-dependent capability demonstration
Evidence freshness	Typically predefined before verification	Statistics obtained during evaluation	Computation-specific proof generated for verification	Fresh, previously undisclosed challenges
Possession-related conclusion	Not explicitly evaluated	Not explicitly evaluated	May bind the demonstrated computation to a committed model	Empirical inference of possession of a matching model under scope
Verifier access to claimant model	Not required	Often required through API or white-box access	Not required	Not required
Dedicated participation by deployed service	Not generally required beyond ordinary responses	Not generally required, but comparison access is needed	Typically required from the model-holding service	Not required beyond ordinary black-box inference

condition and assess whether the claimant's identity claim is technically supported.

III. PROPOSED FRAMEWORK

A. Motivation and Positioning

TP-CRIV is designed to provide an independent third party with empirical evidence about the relationship between a claimant's locally possessed model and a suspicious remotely deployed model when direct model comparison is unavailable. Through fresh challenge-response interactions, the framework enables the verifier to assess whether the claimant demonstrates behavior consistent with the identity specified for the intended verification objective, without requiring access to the claimant's model or protocol-specific cooperation from the deployed service. To make this inference well defined, TP-CRIV separates four concepts that must not be conflated. First, the verification objective specifies an a priori distinction between matching and non-matching operating situations. Second, a model-dependent property supplies information for constructing discriminative requirements but does not define model identity. Third, fresh property-demanding challenges elicit observable challenge-solving capability rather than directly revealing or comparing the property. Fourth, calibration on independent reference configurations provides an operational interpretation of the resulting scores relative to the predeclared identity distinction.

These roles are formalized below. Compared with model-specific evidence approaches, TP-CRIV evaluates not only evidence associated with the claimant and the suspicious service, but also whether the claimant can use a locally available model to satisfy fresh model-dependent requirements. In contrast to model-similarity verification, TP-CRIV does not require the verifier to access or query the claimant's model. Unlike proof-based verifiable inference, it also does not require the suspicious MLaaS provider to participate in a dedicated verification procedure. Table I summarizes these differences from existing verification approaches.

B. System Model

We consider three entities:

- **Prover** $\mathcal{P}$, which locally possesses a candidate model M and claims that M corresponds to the matching situation relative to M^{cloud} .
- **Service provider** $\mathcal{S}$, which operates M^{cloud} through an MLaaS platform.
- **Verifier** $\mathcal{V}$, which specifies a verification objective $\mathcal{O}$, prepares or obtains calibration data, conducts verification, and evaluates the resulting evidence.

Let F^{MLaaS} denote the black-box MLaaS interface exposed by $\mathcal{S}$. For an input x , $\mathcal{V}$ observes only

$$y^{\text{MLaaS}} = F^{\text{MLaaS}}(x), \quad (1)$$

where, in the baseline framework setting,

$$F^{\text{MLaaS}}(x) = M^{\text{cloud}}(x). \quad (2)$$

Neither $\mathcal{P}$ nor $\mathcal{V}$ directly accesses M^{cloud} . Moreover, $\mathcal{V}$ has no direct access to M and need not observe its parameters, architecture, or other internal information. Consequently, $\mathcal{V}$ cannot directly compare the internal structures of M and M^{cloud} . TP-CRIV therefore evaluates the claimed relationship from observable interactions with the deployed MLaaS under the declared verification objective, calibration procedure, and operational assumptions.

C. Verification Scope and Operational Assumptions

a) *Non-Matching Model Configurations*: A model configuration declared non-matching for $\mathcal{O}$ constitutes the negative identity case. Depending on the intended objective, M may be unrelated to, independently trained from, extracted from [3], distilled from [32], or otherwise derived from M^{cloud} . Such provenance does not determine the identity label; the label is fixed by the operating situations declared for $\mathcal{O}$. Acceptance of a declared non-matching configuration constitutes a false acceptance.

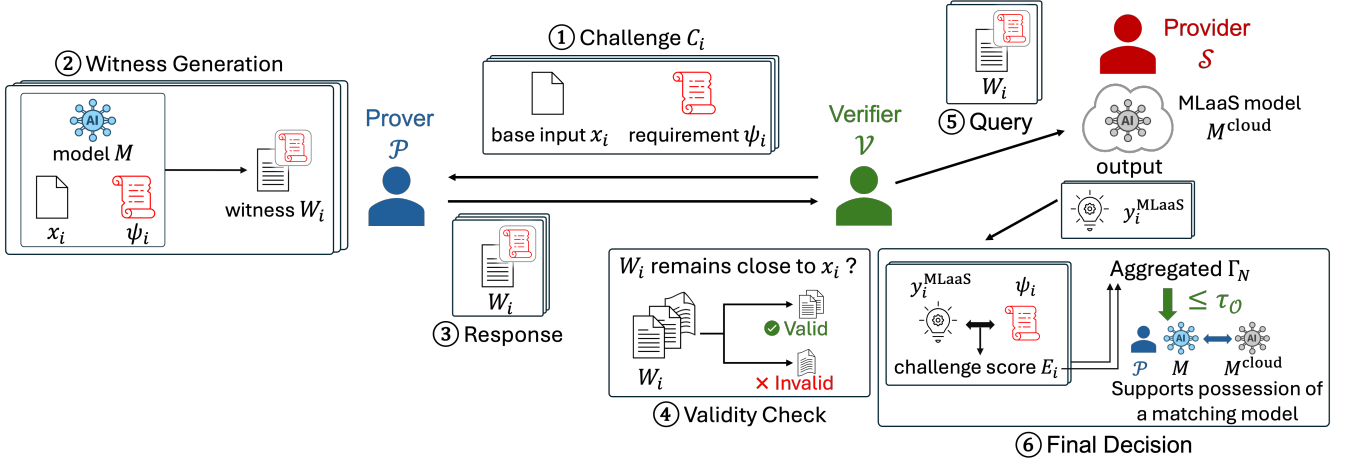


Fig. 2. Overview of the TP-CRIV framework. The verifier specifies the objective-dependent model-identity distinction, selects a model-dependent property, issues fresh property-demanding challenges, and evaluates the prover's returned responses through the deployed MLaaS. The prover answers the challenges using a locally available candidate model.

Model configurations outside the declared operating population are outside the characterized identity scope. Accordingly, the reported verification performance does not automatically extend to model configurations that are not represented in the corresponding calibration, evaluation, or supporting analysis.

b) *Service-Side Assumption*: $\mathcal{S}$ does not disclose M^{cloud} , its parameters, architecture, or inference pipeline. $\mathcal{V}$ interacts with F^{MLaaS} only through its ordinary black-box interface. $\mathcal{S}$ is assumed to return the ordinary output of a fixed deployed inference pipeline for each submitted input. Adaptive routing of verification queries, deliberate output manipulation, and query refusal are outside the present scope. The verification decision therefore concerns the observed behavior of the deployed service and does not establish its internal implementation.

c) *Network Isolation*: Before the first challenge of a verification session is disclosed, $\mathcal{P}$ is placed under network isolation. The isolation remains in effect until all responses for the session have been returned. During this interval, $\mathcal{P}$ may use the disclosed challenges, its locally available model M , and local computational resources, but may not obtain additional challenge-relevant information from F^{MLaaS} , another remote model, a relay service, an external oracle, or any other external source. Thus, network isolation excludes online external assistance after challenge disclosure.

D. TP-CRIV Protocol

A verification trial consists of N challenge-response interactions. For each interaction i , $\mathcal{V}$ issues a fresh challenge, $\mathcal{P}$ generates a corresponding response using its locally available model, and $\mathcal{V}$ evaluates the response through the deployed MLaaS.

Let ResGen denote the response-generation algorithm used by $\mathcal{P}$. The present framework adopts a model-based response-generation scope: ResGen may use M , the disclosed challenge, and local computation required to operate or process M . A standalone learned or precomputed mechanism that can generate responses without using M is outside the present model-based response-generation scope.

Fig. 2 illustrates the overall TP-CRIV framework. The i -th interaction proceeds as follows.

- 1) $\mathcal{V}$ generates a fresh challenge

$$C_i = (x_i, \psi_i), \quad (3)$$

where x_i is a base input and ψ_i specifies the verification requirement.

- 2) Given C_i , $\mathcal{P}$ applies ResGen to its local model M and generates a witness

$$W_i \leftarrow \text{ResGen}(M, C_i). \quad (4)$$

- 3) $\mathcal{P}$ then returns W_i to $\mathcal{V}$.

- 4) $\mathcal{V}$ evaluates the challenge-specific witness validity

$$\text{Valid}_{\text{wit}}(C_i, W_i) \in \{0, 1\}. \quad (5)$$

An invalid witness is excluded from subsequent score evaluation.

- 5) For each valid W_i , $\mathcal{V}$ submits W_i to the MLaaS and obtains

$$y_i^{\text{MLaaS}} = F^{\text{MLaaS}}(W_i). \quad (6)$$

- 6) After the N interactions, $\mathcal{V}$ defines

$$\mathcal{I}_{\text{val}} = \{i \mid \text{Valid}_{\text{wit}}(C_i, W_i) = 1\}, \quad (7)$$

and

$$N_{\text{val}} = |\mathcal{I}_{\text{val}}|. \quad (8)$$

Let $N_{\min}$ denote the minimum number of valid witnesses required for score aggregation. If $N_{\text{val}} < N_{\min}$, the session is rejected. Otherwise, for each $i \in \mathcal{I}_{\text{val}}$, $\mathcal{V}$ computes

$$E_i = \text{Eval}(C_i, W_i, y_i^{\text{MLaaS}}), \quad (9)$$

where Eval denotes the challenge-specific scoring function, and E_i is the resulting per-challenge score.

Let Aggr denote the declared aggregation rule. The valid scores are aggregated as

$$\Gamma_N = \text{Aggr}(\{E_i\}_{i \in \mathcal{I}_{\text{val}}}). \quad (10)$$

Let $\tau_{\mathcal{O}}$ denote the operational acceptance threshold fixed for $\mathcal{O}$ before the suspicious configuration is evaluated. The final decision is

$$\text{Accept} = \begin{cases} 0, & N_{\text{val}} < N_{\text{min}}, \\ \mathbb{I}[\Gamma_N \leq \tau_{\mathcal{O}}], & N_{\text{val}} \geq N_{\text{min}}. \end{cases} \quad (11)$$

$\tau_{\mathcal{O}}$ is calibrated using data prepared independently of the suspicious configuration. Here, $\mathbb{I}[\cdot]$ denotes the indicator function, which returns 1 if the enclosed condition is true and 0 otherwise. Its reported operating characteristics apply only to the model population, challenge distribution, response-generation conditions, and deployment conditions represented in the supporting calibration and evaluation.

In particular, different choices of ResGen may induce different score distributions for the same (M, M^{cloud}) configuration. Accordingly, a reported soundness or false-acceptance characterization covers only the choices of ResGen represented in the corresponding evaluation or supported by separate analysis. Standalone non-model response mechanisms remain outside the present model-based soundness scope.

The interpretation of acceptance under freshness, network isolation, and the stated model-based scope is developed in Section III-E.

E. Verification Principle

TP-CRIV separates four logically distinct elements: the objective-dependent identity distinction, the model-dependent property used to construct challenges, the operational characterization of challenge-solving performance, and the inference obtained by combining matching consistency with the freshness and network-isolation conditions. This separation prevents the verification outcome from defining the identity relation that it is intended to test.

1) *Operating Situations and Objective-Dependent Identity:* For each intended verification problem, $\mathcal{V}$ specifies two disjoint operating situations, $\Pi_1^{\mathcal{O}}$ and $\Pi_0^{\mathcal{O}}$, representing configurations regarded as matching and non-matching, respectively. These situations are specified a priori according to the semantic distinction relevant to the intended application and independently of the challenge-response score, calibration threshold, or acceptance outcome. Together, they define the model-identity distinction tested by the verification objective $\mathcal{O}$.

The corresponding operating population is

$$\Pi^{\mathcal{O}} = \Pi_1^{\mathcal{O}} \cup \Pi_0^{\mathcal{O}}, \quad \Pi_1^{\mathcal{O}} \cap \Pi_0^{\mathcal{O}} = \emptyset. \quad (12)$$

Thus, model identity is objective dependent and population scoped. TP-CRIV does not posit a universal identity relation over all possible AI models.

For convenience, membership in these two situations is encoded by

$$\text{Match}_{\mathcal{O}}(M, M^{\text{cloud}}) = \begin{cases} 1, & (M, M^{\text{cloud}}) \in \Pi_1^{\mathcal{O}}, \\ 0, & (M, M^{\text{cloud}}) \in \Pi_0^{\mathcal{O}}. \end{cases} \quad (13)$$

Here, $\text{Match}_{\mathcal{O}}$ introduces no additional notion of identity and is not inferred from the TP-CRIV score. It only records membership in the two situations specified independently

for $\mathcal{O}$. For example, one verification objective may declare same-model-instance configurations as matching and selected distinct model instances as non-matching. Within the corresponding $\Pi^{\mathcal{O}}$, this constitutes an exact-instance distinction. Another objective may place designated descendant models in the matching situation and selected outside-lineage models in the non-matching situation, yielding a lineage-oriented distinction. The meaning of identity therefore changes when the declared operating situations change.

A configuration outside $\Pi^{\mathcal{O}}$ is outside the identity distinction instantiated by $\mathcal{O}$. The protocol may still produce an observable score for such a configuration, but the current objective does not assign it a matching or non-matching label, and the reported operating characteristics do not automatically apply to it. Extending the identity claim to such a configuration requires extending the declared operating situations and establishing corresponding calibration, evaluation, or analysis.

The choice of ResGen does not define model identity. The matching and non-matching labels are fixed a priori by $\Pi_1^{\mathcal{O}}$ and $\Pi_0^{\mathcal{O}}$. Different choices of ResGen may affect observable challenge-solving performance and operating error rates, but they do not change those predeclared identity labels.

2) *Model-Dependent Property and Challenge-Solving Capability:* For each model M , let S_M denote a selected model-dependent property. Examples include local decision geometry in a classifier, conditional generation behavior in a generative model, or another high-dimensional behavioral structure whose detailed form depends on the model. The selected property does not define $\Pi_1^{\mathcal{O}}$ or $\Pi_0^{\mathcal{O}}$. Instead, $\mathcal{V}$ uses it as a source of model-dependent information from which fresh verification requirements can be constructed.

For $\Pi^{\mathcal{O}}$, a useful property is one whose challenge-relevant information enables configurations in $\Pi_1^{\mathcal{O}}$ to solve requirements determined by M^{cloud} more consistently than configurations in $\Pi_0^{\mathcal{O}}$ under the covered choices of ResGen. For a challenge $C = (x, \psi)$, the requirement ψ is constructed so that a valid low-error solution depends substantially on challenge-relevant aspects of $S_{M^{\text{cloud}}}$. Such a challenge is referred to as a property-demanding challenge.

For each valid witness W , $\mathcal{V}$ evaluates the observable score $\text{Eval}(C, W, F^{\text{MLaaS}}(W))$, which measures how well the returned witness satisfies the issued requirement. $\mathcal{V}$ need not reconstruct S_M , $S_{M^{\text{cloud}}}$, or the complete set of successful witnesses. It observes only whether $\mathcal{P}$ can use M through ResGen to generate successful responses to newly issued challenges. This capability relation is not deterministic. A configuration in $\Pi_1^{\mathcal{O}}$ may occasionally fail because of numerical, algorithmic, or deployment-side effects. Conversely, a configuration in $\Pi_0^{\mathcal{O}}$ may reproduce sufficient challenge-relevant information to solve some or all challenges. Such success does not change its predeclared class membership. If it is accepted, the event constitutes a false acceptance under the declared objective.

Accordingly, the selected property provides the mechanism through which model-dependent information is elicited as an observable challenge-solving capability. Whether this capability actually discriminates the two declared situations must be

established operationally rather than assumed from a converse relation between property similarity and model identity.

3) *Operational Characterization of Challenge-Solving Performance*: Let $\mathcal{D}_C$ denote the declared distribution from which fresh challenges are drawn, and let ResGen denote the choice covered by the operating characterization. For each $b \in \{0, 1\}$, let $Q_b^\mathcal{O}$ denote the declared distribution over model configurations in $\Pi_b^\mathcal{O}$ used for operational characterization. Sampling

$$(M, M^{\text{cloud}}) \sim Q_b^\mathcal{O}, \quad (14)$$

drawing fresh challenges from $\mathcal{D}_C$, generating responses using ResGen, and applying Eq. (10) induce the observable aggregate-score distribution

$$\Gamma_N \sim P_{b, \text{ResGen}}^\mathcal{O}. \quad (15)$$

These distributions provide an operational characterization of the challenge-solving capability demonstrated by the matching and non-matching populations under the specified conditions. The purpose of the selected property and challenge family is therefore to induce sufficiently distinguishable observable performance between $\Pi_1^\mathcal{O}$ and $\Pi_0^\mathcal{O}$.

TP-CRIV does not assume a deterministic, one-to-one, or monotone mapping between a latent discrepancy between S_M and $S_{M^{\text{cloud}}}$ and the observed aggregate score Γ_N . Different model configurations may produce similar scores, the same configuration may produce different scores for different sets of fresh challenges, and different choices of ResGen may induce different score distributions for the same model configuration. The relevant question is instead whether the challenge-solving performance induced by the declared matching and non-matching populations is sufficiently separated for the intended operating criterion.

Using independent calibration data, $\mathcal{V}$ determines $\tau_\mathcal{O}$ before evaluating the suspicious configuration. Let RG_0 and RG_1 denote the covered choices of ResGen for the non-matching and matching situations, respectively; they may coincide. The corresponding false-acceptance rate ($\text{FAR}_{\mathcal{O}, RG_0}$) and false-rejection rate ($\text{FRR}_{\mathcal{O}, RG_1}$) are given as follows:

$$\text{FAR}_{\mathcal{O}, RG_0} = \Pr_{\substack{(M, M^{\text{cloud}}) \sim Q_0^\mathcal{O} \\ C_{1:N} \sim \mathcal{D}_C^N}} [\Gamma_N \leq \tau_\mathcal{O} \mid RG_0], \quad (16)$$

and

$$\text{FRR}_{\mathcal{O}, RG_1} = \Pr_{\substack{(M, M^{\text{cloud}}) \sim Q_1^\mathcal{O} \\ C_{1:N} \sim \mathcal{D}_C^N}} [\Gamma_N > \tau_\mathcal{O} \mid RG_1]. \quad (17)$$

The probabilities include the randomness of model configuration sampling, fresh challenge generation, and, when applicable, the response-generation algorithm. $\tau_\mathcal{O}$ provides an operational boundary for interpreting observed challenge-solving performance relative to the declared $\Pi_1^\mathcal{O}$ and $\Pi_0^\mathcal{O}$ situations.

Within the declared operating population, acceptance provides finite-sample empirical evidence that the observed challenge-solving performance is more consistent with the characterized matching population than with the characterized non-matching population. TP-CRIV evaluates evidence for

TABLE II
VERIFIER'S INFERENCE PROCESS IN TP-CRIV.

Step	Verifier's observation or inference
1	$\mathcal{P}$ repeatedly returns valid low-score witnesses for fresh property-demanding challenges.
↓	<i>The transcript directly demonstrates finite challenge-solving capability.</i>
2	Freshness limits replay-based strategies, while network isolation excludes online external assistance after challenge disclosure.
↓	<i>$\mathcal{P}$ locally possesses the challenge-solving capability required to generate the demonstrated responses without online external assistance.</i>
3	Under the model-based response-generation scope, the aggregate score lies in the acceptance region calibrated for the declared matching situation.
↓	<i>The candidate model exhibits matching-consistent challenge-solving performance within the calibrated scope.</i>
4	The combined evidence supports an empirical inference that the claimant possesses a model satisfying the objective-dependent identity.

membership in a predeclared identity class rather than defining identity from acceptance.

The reported $\tau_\mathcal{O}$, FAR, and FRR apply only to the declared operating distributions, $\mathcal{D}_C$, ResGen, and deployment conditions. Broader claims require corresponding extension of the operating characterization.

4) Inference Under Freshness and Network Isolation:

The calibrated decision characterizes whether the observed challenge-solving performance is consistent with the declared matching situation; by itself, it does not establish whether the response was generated without external assistance. Fresh challenges limit replay based on a small precomputed response set, while the network-isolation condition prevents $\mathcal{P}$ from obtaining additional challenge-relevant information from external sources after disclosure.

Within the stated model-based response-generation scope, the session responses are generated through $\text{ResGen}(M, C)$. If Γ_N also lies in the calibrated matching region, the observed capability is characteristic of the matching situation within the declared operating scope. These conditions jointly provide an empirical basis for inferring that $\mathcal{P}$ possesses a model satisfying the identity specified by $\mathcal{O}$. The inference is conditional on the stated scope and does not establish cryptographic binding, training provenance, or legal ownership.

The verifier's inference process is summarized in Table II.

F. Protocol Requirements

A valid TP-CRIV instantiation should satisfy the following requirements under Eq. (11).

a) *Objective and Population Specification*: $\mathcal{V}$ must specify the disjoint situations $\Pi_1^\mathcal{O}$ and $\Pi_0^\mathcal{O}$ before evaluating the suspicious case, independently of the TP-CRIV score, $\tau_\mathcal{O}$, or the eventual decision. $\mathcal{V}$ must also specify the operating distributions $Q_1^\mathcal{O}$ and $Q_0^\mathcal{O}$, $\mathcal{D}_C$, the choices of ResGen covered by calibration and evaluation, and the criterion used to select $\tau_\mathcal{O}$. $\Pi_0^\mathcal{O}$ should contain the model configurations that constitute relevant alternatives to the intended identity claim; omitted

model classes and unrepresented algorithms remain outside the characterized soundness scope.

b) *Completeness*: Let ϵ_c denote the allowed completeness error. For $(M, M^{\text{cloud}}) \sim Q_1^{\mathcal{O}}$ and a covered ResGen, acceptance should occur with high probability:

$$\Pr_{\substack{(M, M^{\text{cloud}}) \sim Q_1^{\mathcal{O}} \\ C_{1:N} \sim \mathcal{D}_C^N}} [\text{Accept} = 1 \mid \text{ResGen}] \geq 1 - \epsilon_c. \quad (18)$$

Here, ϵ_c is the objective-specific false-rejection probability under the declared operating conditions.

c) *Soundness*: Let ϵ_s denote the allowed soundness error. For $(M, M^{\text{cloud}}) \sim Q_0^{\mathcal{O}}$ and a covered ResGen, acceptance should occur with low probability:

$$\Pr_{\substack{(M, M^{\text{cloud}}) \sim Q_0^{\mathcal{O}} \\ C_{1:N} \sim \mathcal{D}_C^N}} [\text{Accept} = 1 \mid \text{ResGen}] \leq \epsilon_s. \quad (19)$$

Here, ϵ_s is the objective-specific false-acceptance probability under the declared operating conditions. The identity label is independent of ResGen; therefore, acceptance of any configuration assigned to $\Pi_0^{\mathcal{O}}$ is a false acceptance.

d) *Property-Demanding Challenge Design*: Each challenge $C = (x, \psi)$ must contain a fresh requirement whose successful solution depends substantially on model-dependent information associated with $S_{M^{\text{cloud}}}$. The challenge family should separate the induced score distributions of the relevant matching and non-matching populations at the intended operating point. A fixed witness, a small precomputed response set, or challenge-independent evidence should not yield acceptance with high probability over fresh challenges.

e) *Calibration and Threshold Transfer*: $\tau_{\mathcal{O}}$ must be determined from calibration data prepared independently of the suspicious configuration and fixed before that configuration is evaluated. Reported error rates are valid only under the challenge-generation process, validity conditions, aggregation rule, ResGen, and deployment pipeline represented in calibration and evaluation.

f) *Freshness and Network Isolation*: Each $C_i \sim \mathcal{D}_C$ must remain unpredictable before disclosure, and the challenge family should limit reuse of responses across fresh challenges. The network-isolation condition defined above must hold throughout witness generation. Both conditions are required for the inference of Section III-E.

IV. CNN-BASED INSTANTIATION AND FEASIBILITY EVALUATION

This section instantiates the abstract TP-CRIV components for CNN-based image classification under an explicitly bounded experimental setting. The evaluation fixes ResGen = I-FMPC for all tested (M, M^{cloud}) configurations and examines whether the declared matching and non-matching situations induce separable scores and whether a threshold calibrated on separate models transfers to held-out configurations. The results characterize only the operating scope instantiated below.

A. Verification Objective and Operating Scope

We consider a CNN-based feasibility demonstration of TP-CRIV under an explicitly bounded operating scope. Both M and M^{cloud} are CNN image classifiers. Let $\mathcal{M}_{\text{eval}}$ denote the finite model pool consisting of ten ImageNet-pretrained CNN model instances: ResNet-18, -34, -101, and -152; VGG-11, -13, -16, and -19; and two ResNet-50 instances using the TorchVision IMAGENET1K_V1 and IMAGENET1K_V2 pretrained weights, respectively [33]–[35]. The two ResNet-50 instances are hereafter denoted by ResNet-50 (α) and ResNet-50 (β). They share the same architecture but contain different pretrained parameter sets obtained using different training recipes. The evaluated pool is a heterogeneous set of publicly available pretrained model instances that differ primarily in architecture family, network depth, or pretrained weight configuration. All models perform the same ImageNet classification task and share the same output-label space.

For the present feasibility study, the operating population is declared as

$$\Pi^{\mathcal{O}} = \mathcal{M}_{\text{eval}} \times \mathcal{M}_{\text{eval}}. \quad (20)$$

Thus, $\Pi^{\mathcal{O}}$ contains all directed (M, M^{cloud}) configurations formed from the ten models in $\mathcal{M}_{\text{eval}}$. Before challenge-response evaluation, the verification objective $\mathcal{O}$ declares the following two situations within $\Pi^{\mathcal{O}}$. These declarations are independent of the observed scores and thresholds.

a) *Matching situation*: The first situation is

$$\Pi_1^{\mathcal{O}} = \{(M, M^{\text{cloud}}) \in \Pi^{\mathcal{O}} \mid M = M^{\text{cloud}}\}. \quad (21)$$

It contains the 10 same-model directed configurations, which are designated as the identity-satisfying, or matching, situation for the present objective.

b) *Non-matching situation*: The contrasting situation is

$$\Pi_0^{\mathcal{O}} = \{(M, M^{\text{cloud}}) \in \Pi^{\mathcal{O}} \mid M \neq M^{\text{cloud}}\}. \quad (22)$$

It contains the 90 cross-model directed configurations formed from distinct model instances in $\mathcal{M}_{\text{eval}}$, which are designated as the non-matching situation. For the finite operating characterization below, $Q_1^{\mathcal{O}}$ and $Q_0^{\mathcal{O}}$ are taken as uniform distributions over the 10 and 90 configurations, respectively.

In the present feasibility study, both matching and non-matching provers use ResGen = I-FMPC. Thus, the reported score distributions and FAR characterize non-matching models that attempt the same fine-grained probability-control task as matching models. Other model-based choices of ResGen are outside the present operating characterization.

Together, these situations define an exact-instance distinction scoped to $\Pi^{\mathcal{O}}$. Models outside $\mathcal{M}_{\text{eval}}$, including independently trained, fine-tuned [36], pruned [37], quantized [38], extracted, or distilled models, are not assigned a label by this experimental objective and are not characterized by the reported FAR. Each evaluated M is used directly with the fixed ResGen specified above.

B. Verification Procedure

Fig. 3 illustrates the verification procedure. The selected property S_M is the local input–probability geometry induced

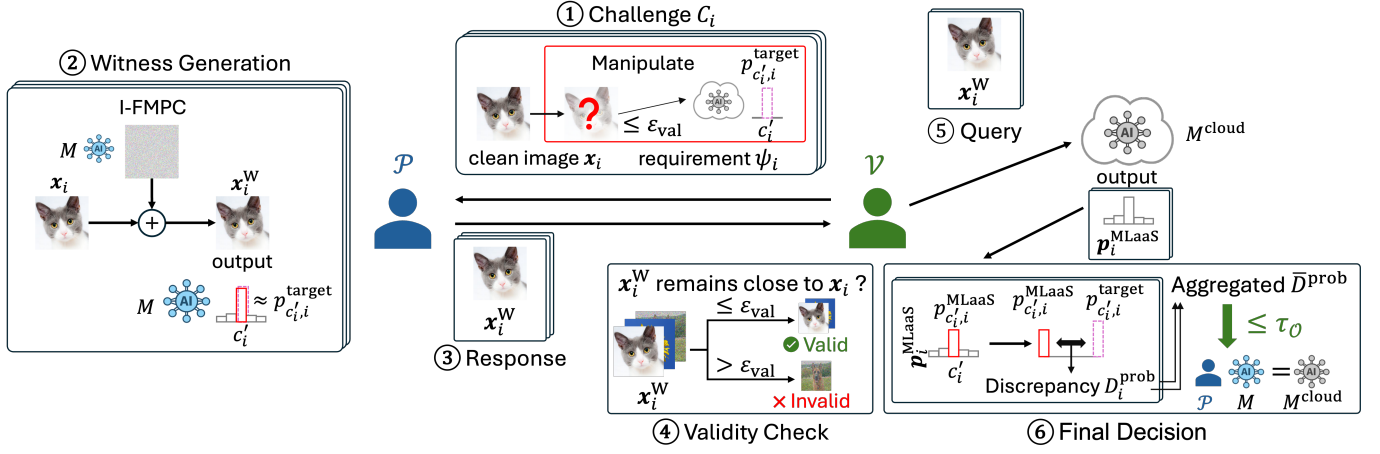


Fig. 3. Verification procedure in the image-classification instantiation of TP-CRIV. $\mathcal{V}$ specifies an input image, a fresh probability-control requirement, and a witness-modification constraint.

by CNN model M , namely the local relationship between bounded input modifications and the resulting class probability changes. This property guides the construction of challenges whose valid low-error solutions depend on $S_{M_{\text{cloud}}}$.

For each verification interaction, $\mathcal{V}$ generates a challenge

$$C = (\mathbf{x}, \psi), \quad (23)$$

where $\mathbf{x}$ is a clean input image and

$$\psi = (c', p_{c'}^{\text{target}}, \epsilon_{\text{val}}). \quad (24)$$

Let $\mathcal{Y}$ denote the output-label space. Here, $\mathcal{V}$ selects the target class $c' \in \mathcal{Y}$ and specifies the required output probability $p_{c'}^{\text{target}}$ and the maximum permitted witness modification ϵ_{val} . The $\mathcal{D}_C$ used in the experiments is instantiated as follows. $\mathbf{x}$ is randomly selected from the ImageNet dataset [39], c' is randomly selected from the full set of 1000 ImageNet classes, and $p_{c'}^{\text{target}}$ is randomly selected from $\{0.10, 0.15, \dots, 0.40\}$. The witness-validity tolerance is fixed at $\epsilon_{\text{val}} = 4/255$. Fresh base inputs are not reused within a verification session.

For each session, $\mathcal{V}$ draws N fresh challenges according to this distribution, with base inputs sampled without replacement within the session:

$$C_i \sim \mathcal{D}_C, \quad i = 1, \dots, N, \quad (25)$$

where

$$C_i = (\mathbf{x}_i, \psi_i) \quad (26)$$

and

$$\psi_i = (c'_i, p_{c'_i,i}^{\text{target}}, \epsilon_{\text{val}}). \quad (27)$$

$\mathbf{x}_i$, c'_i , and $p_{c'_i,i}^{\text{target}}$ are not disclosed before the corresponding challenge is issued.

The ResGen evaluated in the main experiments is instantiated by the Iterative Fine-grained Multi-class Probability Control (I-FMPC) procedure, detailed in Section IV-D. Given C_i , $\mathcal{P}$ applies $\text{ResGen} = \text{I-FMPC}$ to its local model M :

$$\mathbf{x}_i^W = \text{ResGen}(M, C_i) = \text{I-FMPC}(M, C_i). \quad (28)$$

ResGen is used for both Π_1^O and Π_0^O in the reported main experiments.

After receiving $\mathbf{x}_i^W$ from $\mathcal{P}$, $\mathcal{V}$ first checks

$$\|\mathbf{x}_i^W - \mathbf{x}_i\|_{\infty} \leq \epsilon_{\text{val}}. \quad (29)$$

Thus,

$$\text{Valid}_{\text{wit}}(C_i, \mathbf{x}_i^W) = \mathbb{I}[\|\mathbf{x}_i^W - \mathbf{x}_i\|_{\infty} \leq \epsilon_{\text{val}}]. \quad (30)$$

The valid-response index set and count are

$$\mathcal{I}_{\text{val}} = \{i \in \{1, \dots, N\} \mid \text{Valid}_{\text{wit}}(C_i, \mathbf{x}_i^W) = 1\}, \quad (31)$$

$$N_{\text{val}} = |\mathcal{I}_{\text{val}}|.$$

If $N_{\text{val}} < N_{\text{min}}$, $\mathcal{V}$ rejects the session without computing an aggregate score. Otherwise, for each valid witness, $\mathcal{V}$ obtains

$$p_i^{\text{MLaaS}} = F^{\text{MLaaS}}(\mathbf{x}_i^W) \quad (32)$$

and defines the output probability of class c'_i as

$$p_{c'_i,i}^{\text{MLaaS}} = F^{\text{MLaaS}}(\mathbf{x}_i^W)[c'_i]. \quad (33)$$

The per-challenge score is

$$E_i = D_i^{\text{prob}} = \frac{|p_{c'_i,i}^{\text{target}} - p_{c'_i,i}^{\text{MLaaS}}|}{p_{c'_i,i}^{\text{target}}}. \quad (34)$$

For $N_{\text{val}} \geq N_{\text{min}}$, the aggregate score is

$$\Gamma_N = \bar{D}^{\text{prob}} = \frac{1}{N_{\text{val}}} \sum_{i \in \mathcal{I}_{\text{val}}} D_i^{\text{prob}}. \quad (35)$$

The final decision rule is

$$\text{Accept} = \begin{cases} 0, & N_{\text{val}} < N_{\text{min}}, \\ \mathbb{I}[\bar{D}^{\text{prob}} \leq \tau_O], & N_{\text{val}} \geq N_{\text{min}}. \end{cases} \quad (36)$$

Invalid witnesses are excluded from the aggregate, whereas N_{min} prevents acceptance based only on a favorable subset of challenges. τ_O is calibrated from reference configurations drawn from the instantiated operating populations independently of the configuration under evaluation.

Table III summarizes the CNN instantiation of TP-CRIV.

TABLE III
INSTANTIATION OF THE PROPOSED TP-CRIV.

Component	Instantiation
$\mathcal{O}$	Population-scoped exact-instance distinction within $\Pi^\mathcal{O}$
ResGen	Response-generation algorithm I-FMPC
$\Pi_1^\mathcal{O}$	Same-model directed configurations in $\mathcal{M}_{\text{eval}}$
$\Pi_0^\mathcal{O}$	Cross-model directed configurations in $\mathcal{M}_{\text{eval}}$
M	Prover's CNN image classifier
M^{cloud}	Deployed MLaaS CNN image classifier
F^{MLaaS}	$F^{\text{MLaaS}}(\mathbf{x}) = M^{\text{cloud}}(\mathbf{x})$
S	Local input–probability geometry
$N_{\min}$	N in the reported CNN evaluation
$\mathcal{D}_C$	ImageNet-based fresh probability-control challenges
ψ	$(c', p_{c'}^{\text{target}}, \varepsilon_{\text{val}})$
W	$\mathbf{x}^W$
Valid _{wit}	$\mathbb{I}[\ \mathbf{W} - \mathbf{x}\ _\infty \leq \varepsilon_{\text{val}}]$
Eval	$\overline{D}^{\text{prob}}$
Aggr	Arithmetic mean over valid per-challenge scores
Γ_N	$\overline{D}^{\text{prob}}$
Accept	Reject if $N_{\text{val}} < N_{\min}$; otherwise accept iff $\overline{D}^{\text{prob}} \leq \tau_\mathcal{O}$

C. Verification Principle

For the verification objective defined above, the selected model-dependent property should provide challenge-relevant information that induces distinguishable challenge-solving behavior between $\Pi_1^\mathcal{O}$ and $\Pi_0^\mathcal{O}$. In particular, under the instantiated challenge distribution and response-generation algorithm, configurations in $\Pi_1^\mathcal{O}$ should satisfy requirements determined by M^{cloud} more consistently than configurations in $\Pi_0^\mathcal{O}$.

For this purpose, we select S as the local input–probability geometry, namely the relationship between bounded input modifications and the resulting class-probability changes. Behavior near decision boundaries has been used as model-sensitive information in prior fingerprinting studies [13]–[16], which motivates its use here as a property from which discriminative verification requirements can be constructed.

Based on this property, $\mathcal{V}$ issues fresh probability-control challenges. For a challenge $C = (\mathbf{x}, c', p_{c'}^{\text{target}}, \varepsilon_{\text{val}})$, $\mathcal{P}$ must generate a witness satisfying

$$\|\mathbf{x}^W - \mathbf{x}\|_\infty \leq \varepsilon_{\text{val}} \quad (37)$$

and

$$F^{\text{MLaaS}}(\mathbf{x}^W)[c'] \approx p_{c'}^{\text{target}}. \quad (38)$$

The corresponding solution region depends on $S_{M^{\text{cloud}}}$, whereas $\mathcal{P}$ searches for a solution using S_M through ResGen. Varying $\mathbf{x}$, c' , and $p_{c'}^{\text{target}}$ therefore probes different local regions and probability levels.

For $(M, M^{\text{cloud}}) \in \Pi_1^\mathcal{O}$, the same model determines both the search and evaluation geometry. In contrast, for $(M, M^{\text{cloud}}) \in \Pi_0^\mathcal{O}$, a witness satisfying the target probability on M need not lie in the corresponding solution region of M^{cloud} . The procedure therefore induces the observable score distributions $P_{1, \text{ResGen}}^\mathcal{O}$ and $P_{0, \text{ResGen}}^\mathcal{O}$ under the instantiated $\mathcal{D}_C$ and ResGen. The experiments below evaluate whether these distributions are sufficiently separated for threshold calibration and finite-challenge decisions.

D. Witness Generation Algorithm: I-FMPC

For the ResGen characterized in the main experiments, we use I-FMPC. In the CNN-based instantiation of TP-CRIV, the selected property S is the local input–probability geometry. I-FMPC uses white-box information from M to search for a solution to each fresh probability-control challenge. Gradient information has been widely utilized in white-box adversarial-example generation [40]–[42] as an efficient mechanism for identifying local output-change directions. I-FMPC uses this information to navigate the local probability geometry of M and locate a witness that reaches the probability level designated by $\mathcal{V}$ within the permitted neighborhood.

I-FMPC is designed for fine-grained probability control rather than misclassification-oriented optimization. It adjusts each target-class probability toward a specific value through adaptive balancing of the target-class gradients and the gradient of an auxiliary non-target class. Let

$$\mathcal{T}' = \{c'_j\}_{j=1}^m \subset \mathcal{V} \quad (39)$$

denote the target-class set. The auxiliary class c is then selected as

$$c = \arg \max_{k \in \mathcal{V} \setminus \mathcal{T}'} M(\mathbf{x})[k]. \quad (40)$$

Thus, c is the highest-probability class of M among the classes outside $\mathcal{T}'$. The controlled class set is then defined as

$$\mathcal{T} = \{c\} \cup \mathcal{T}'. \quad (41)$$

For each $c'_j \in \mathcal{T}'$, $\mathcal{V}$ specifies a target probability $p_{c'_j}^{\text{target}}$. We denote the set of target probabilities by

$$\mathbf{p}_{\mathcal{T}'}^{\text{target}} = \{p_{c'_j}^{\text{target}}\}_{j=1}^m. \quad (42)$$

The probabilities of the classes in $\mathcal{T}'$ are controlled toward these specified values. When these probabilities exceed their prescribed ranges, the contribution of the auxiliary class c is increased to counterbalance further increases and help maintain a relatively high probability for c .

Given an input image $\mathbf{x}$, I-FMPC generates a probability-controlled input $\mathbf{x}^W = \text{I-FMPC}(M, C)$ with the primary objective

$$M(\mathbf{x}^W)[c'_j] \approx p_{c'_j}^{\text{target}}, \quad j = 1, \dots, m. \quad (43)$$

In addition, the generation algorithm encourages the probability of the auxiliary class c to remain relatively high during probability control. It should be noted that the auxiliary class c is not included in the verification requirement specified by $\mathcal{V}$. Instead, c is selected and used solely by $\mathcal{P}$ as an internal aid during witness generation.

Let $p_k(\mathbf{x}) = M(\mathbf{x})[k]$ denote the softmax probability assigned to class k . For each $k \in \mathcal{T}$, we define

$$L(\mathbf{x}, k) = -\log p_k(\mathbf{x}). \quad (44)$$

I-FMPC uses the weighted class-wise loss

$$L_{\text{FMPC}}(\mathbf{x}_t^W) = \sum_{k \in \mathcal{T}} \beta^k L(\mathbf{x}_t^W, k), \quad (45)$$

where β^k controls the contribution of class k to the combined input gradient. The target-class terms increase the probabilities

TABLE IV
I-FMPC SETTINGS USED FOR THE REPRESENTATIVE
PROBABILITY-CONTROL TRAJECTORIES IN FIG. 4.

	T^{diff}	ε_{gen}	α^{com}	$\mathbf{p}_{c'}^{\text{target}}$
$m = 1$	1×10^{-3}	4/255	1×10^{-3}	{0.30}
$m = 2$	1×10^{-3}	12/255	5×10^{-3}	{0.30, 0.15}
$m = 3$	3×10^{-3}	12/255	5×10^{-3}	{0.30, 0.15, 0.10}

TABLE V
COMMON HYPERPARAMETER SETTINGS FOR I-FMPC.

Hyperparameter	Setting
Maximum iterations t^{max}	10,000
Averaging interval l	5
Minimum step threshold α_{th}	1×10^{-10}
Step decay factor γ	0.5

of $c'_j \in \mathcal{T}'$, while the term for c counterbalances their excessive increases. The iterative update rule of I-FMPC is given by

$$\begin{aligned} \mathbf{x}_0^{\text{W}} &= \mathbf{x}, \\ \mathbf{x}_{t+1}^{\text{W}} &= \text{Clip}_{\mathbf{x}, \varepsilon_{\text{gen}}} \left\{ \mathbf{x}_t^{\text{W}} - \alpha^{\text{com}} \text{sign} \left(\nabla_{\mathbf{x}} L_{\text{FMPC}}(\mathbf{x}_t^{\text{W}}) \right) \right\} \\ &= \text{Clip}_{\mathbf{x}, \varepsilon_{\text{gen}}} \left\{ \mathbf{x}_t^{\text{W}} - \alpha^{\text{com}} \text{sign} \left(\sum_{k \in \mathcal{T}} \beta^k \nabla_{\mathbf{x}} L(\mathbf{x}_t^{\text{W}}, k) \right) \right\}. \end{aligned} \quad (46)$$

Here, α^{com} denotes the common update step size, and β^i controls the contribution of class i to the combined gradient. It should be noted that the class-wise loss $L(\mathbf{x}, i)$ does not directly contain the target probability $p_{c'_j}^{\text{target}}$. Instead, I-FMPC realizes fine-grained probability matching through the adaptive control mechanism summarized in Algorithm 1. In the algorithm, FMPC($\cdot$) denotes one input-update step defined in Eq. (46). For each target class c'_j , the corresponding weight $\beta^{c'_j}$ is increased when its recent average probability remains below the prescribed target range. Conversely, when at least one target-class probability exceeds its prescribed range while all target-class probabilities remain above their lower bounds, the weight β^c of the auxiliary non-target class is increased to counterbalance the excessive increase in the target-class probabilities and help maintain a relatively high probability for c . Once all target-class probabilities enter the neighborhood of their specified values, the common step size α^{com} is reduced for finer convergence. Thus, I-FMPC achieves probability matching by adaptively balancing the class-wise cross-entropy gradients and refining the update step near the target region, rather than by directly minimizing a probability-distance loss.

In the experiments presented in this paper, we employ the dual-class configuration ($m = 1$). For each target class c' specified by $\mathcal{V}$, $\mathcal{P}$ selects

$$c = \arg \max_{k \in \mathcal{V} \setminus \{c'\}} M(\mathbf{x})[k] \quad (47)$$

as the auxiliary competing class. I-FMPC then controls the probability of c' toward $p_{c'}^{\text{target}}$ while using c to counterbalance excessive changes in the target-class probability. To illustrate the behavior of I-FMPC, Fig. 4 presents representative

Algorithm 1 I-FMPC: Iterative Fine-grained Multi-class Probability Control

Require: Input image $\mathbf{x}$, model M with output-label space $\mathcal{Y}$, target class set $\mathcal{T}' = \{c'_j\}_{j=1}^m$, target probabilities $\mathbf{p}_{\mathcal{T}'}^{\text{target}} = \{p_{c'_j}^{\text{target}}\}_{j=1}^m$, shared step size α^{com} , generation perturbation budget ε_{gen} , averaging interval l , tolerance T^{diff} , maximum iterations t^{max} , minimum step threshold α_{th} , step decay factor $\gamma \in (0, 1)$

Ensure: Probability-controlled input $\mathbf{x}^{\text{W}}$

```

1:  $\mathbf{x}_0^{\text{W}} = \mathbf{x}$ 
2:  $c \leftarrow \arg \max_{k \in \mathcal{Y} \setminus \mathcal{T}'} M(\mathbf{x})[k]$ 
3:  $\mathcal{T} \leftarrow \{c\} \cup \mathcal{T}'$ 
4:  $\mathbf{x}^{\text{W}} = \mathbf{x}_0^{\text{W}}$ 
5:  $\beta^i \leftarrow 1, \quad \forall i \in \mathcal{T}$ 
6: for  $t = 1$  to  $t^{\text{max}}$  do
7:    $\mathbf{x}_t^{\text{W}} = \text{FMPC}(\mathbf{x}_{t-1}^{\text{W}}, \mathbf{x}, M, \alpha^{\text{com}}, \{\beta^i\}_{i \in \mathcal{T}}, \varepsilon_{\text{gen}})$ 
8:    $\mathbf{x}^{\text{W}} \leftarrow \mathbf{x}_t^{\text{W}}$ 
9:    $\mathbf{p}_t = M(\mathbf{x}_t^{\text{W}})$ 
10:  if  $t \bmod l = 0$  then
11:     $\mathbf{p}^{\text{mean}} = \frac{1}{l} \sum_{k=t-l+1}^t \mathbf{p}_k$ 
12:    for each  $j = 1$  to  $m$  do
13:      if  $p_{c'_j}^{\text{mean}} < (1 - T^{\text{diff}}) p_{c'_j}^{\text{target}}$  then
14:         $\beta^{c'_j} \leftarrow \beta^{c'_j} + 1$ 
15:      end if
16:    end for
17:    if  $(\exists j \in \{1, \dots, m\} : p_{c'_j}^{\text{mean}} > (1 + T^{\text{diff}}) p_{c'_j}^{\text{target}})$ 
18:      and
19:       $(\forall j \in \{1, \dots, m\} : p_{c'_j}^{\text{mean}} > (1 - T^{\text{diff}}) p_{c'_j}^{\text{target}})$ 
20:         $\beta^c \leftarrow \beta^c + 1$ 
21:      end if
22:      if  $\max_{j \in \{1, \dots, m\}} \left| \frac{p_{c'_j}^{\text{mean}} - p_{c'_j}^{\text{target}}}{p_{c'_j}^{\text{target}}} \right| \leq T^{\text{diff}}$  then
23:         $\alpha^{\text{com}} \leftarrow \gamma \alpha^{\text{com}}$ 
24:      end if
25:      if  $\alpha^{\text{com}} < \alpha_{\text{th}}$  then
26:        break
27:      end if
28:    end for
29:  return  $\mathbf{x}^{\text{W}}$ 

```

probability trajectories for $m = 1, 2$, and 3. The corresponding configuration-dependent hyperparameter settings are summarized in Table IV, while the common hyperparameter settings used throughout the evaluations are listed in Table V. As shown in Fig. 4, the target-class probabilities approach their specified values, while the auxiliary class provides a counterbalancing contribution during probability control.

By utilizing gradient information derived from M , I-FMPC enables $\mathcal{P}$ to search for a witness satisfying the fresh requirement ψ specified by $\mathcal{V}$. The algorithm therefore realizes ResGen by using S_M to construct a candidate solution. Whether this candidate also satisfies the requirement on M^{cloud} is then evaluated through F^{MLaaS} . Thus, I-FMPC operationalizes the central TP-CRIV principle: information

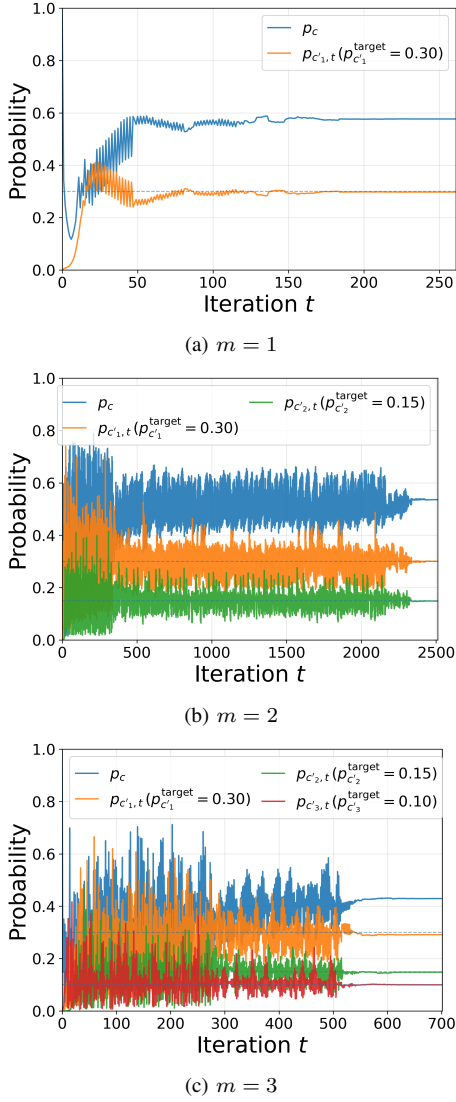


Fig. 4. Trajectory of output probabilities $p_{k,t}$ for the controlled classes $k \in \mathcal{T}$ over iterations t , where $\mathbf{x}_t^W$ denotes the image generated by I-FMPC at iteration t and $p_{k,t} = M(\mathbf{x}_t^W)[k]$. The probabilities of the target classes in $\mathcal{T}'$ converge toward their designated values, while the auxiliary class c selected by $\mathcal{P}$ maintains a relatively high probability throughout the illustrated trajectories.

about the selected property is not compared directly, but is used to solve a challenge whose valid solution is determined by that property.

E. Feasibility Assessment

This subsection empirically characterizes the challenge-induced score behavior of the instantiated operating populations Π_1^O and Π_0^O under $\mathcal{D}_C$ and ResGen specified above. Specifically, we examine whether I-FMPC can solve the property-demanding challenges for configurations in Π_1^O , whether the resulting scores are separated from those obtained for configurations in Π_0^O , and whether a threshold estimated from calibration configurations transfers to held-out configurations within the same operating world.

1) *Experimental Setting*: The evaluation includes all 10 configurations in Π_1^O and all 90 configurations in Π_0^O constructed from $\mathcal{M}_{\text{eval}}$. For both Π_1^O and Π_0^O , ResGen = I-FMPC is fixed. Importantly, the non-matching configurations use the same fine-grained probability-control algorithm and optimize the same target probability specified by $\mathcal{V}$ on their respective models.

$\mathcal{P}$ uses the generation budget

$$\varepsilon_{\text{gen}} = \varepsilon_{\text{val}} = \frac{4}{255}. \quad (48)$$

Witnesses are generated using the $m = 1$ configuration of I-FMPC. The configuration-dependent parameters are reported in Table IV, while the shared stopping and step-control parameters are listed in Table V. The target probability specified by $\mathcal{V}$ is sampled per challenge as described in the verification procedure above.

All witnesses generated in the model-pair and threshold-calibration experiments satisfied

$$\|\mathbf{x}^W - \mathbf{x}\|_{\infty} \leq \varepsilon_{\text{gen}} \leq \varepsilon_{\text{val}}, \quad (49)$$

and I-FMPC reached its stopping condition before the maximum iteration limit. Therefore, all returned witnesses were valid and included in the subsequent score evaluation. Consequently, $N_{\text{val}} = N$ for every verification trial reported below. For this CNN instantiation, we set $N_{\text{min}} = N$, so a session cannot be accepted by returning valid witnesses only for a favorable subset of the issued challenges. To further characterize the generated witnesses, we measured their perceptual similarity to the corresponding base images. Across the ten evaluated models, the mean SSIM [43] ranged from 0.9890 to 0.9933, the mean LPIPS [44] from 0.0024 to 0.0056, and the mean RMSE from 0.0049 to 0.0060. These results indicate that the probability-control requirements are satisfied using only small and perceptually subtle modifications to the base images.

2) *Observable Separation Induced by Property-Demanding Challenges*: To characterize the score behavior induced by Π_1^O and Π_0^O , we generate 100 fresh challenges sampled according to $\mathcal{D}_C$ for each directed configuration in $\Pi_1^O \cup \Pi_0^O$. For every challenge, ResGen = I-FMPC is applied to M . Thus, matching and non-matching configurations use the same probability-control objective and algorithm; only the relationship between the local and deployed models changes.

For each directed model configuration, the per-challenge discrepancies are aggregated as

$$\Gamma_{100} = \overline{D}^{\text{prob}} = \frac{1}{100} \sum_{i=1}^{100} D_i^{\text{prob}}. \quad (50)$$

Fig. 5 shows the resulting $\overline{D}^{\text{prob}}$ for all evaluated configurations.

For configurations in Π_1^O , represented by the diagonal entries, $\overline{D}^{\text{prob}}$ remains close to zero. Because $M = M^{\text{cloud}}$, I-FMPC uses the same local probability information that determines the challenge solution on M^{cloud} . These results therefore demonstrate that the selected challenge family is consistently solvable for Π_1^O under the specified ResGen.

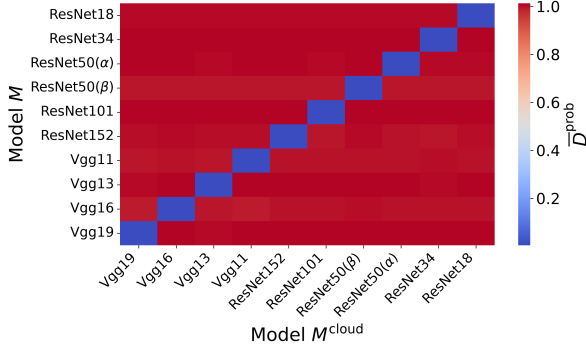


Fig. 5. Heatmap of $\bar{D}^{\text{prob}}$ for the evaluated directed model configurations. The diagonal entries correspond to same-model configurations, whereas the off-diagonal entries correspond to cross-model configurations.

By contrast, all configurations in $\Pi_0^{\mathcal{O}}$, represented by the off-diagonal entries, exhibit substantially larger values of $\bar{D}^{\text{prob}}$. For these configurations, I-FMPC locates a solution with respect to the local input–probability geometry of M , whereas the requirement is evaluated on M^{cloud} . The resulting witness therefore generally fails to reach the probability designated by $\mathcal{V}$ on M^{cloud} .

Thus, under the instantiated $\mathcal{D}_C$ and ResGen, the evaluated property-demanding challenge family induces observable separation between the score behavior of $\Pi_1^{\mathcal{O}}$ and $\Pi_0^{\mathcal{O}}$. The heatmap does not represent a direct distance between complete model decision landscapes. It represents finite-challenge solving error produced by the declared challenge-response procedure.

The 100-challenge experiment characterizes separation within $\Pi^{\mathcal{O}}$ only. The next experiment examines whether this separation supports binary decisions with fewer fresh challenges.

3) Finite-Challenge Verification and Threshold Calibration:

The preceding experiment shows observable separation between the score behavior induced by $\Pi_1^{\mathcal{O}}$ and $\Pi_0^{\mathcal{O}}$ under the instantiated $\mathcal{D}_C$ and ResGen. We next examine whether this separation can be converted into finite-challenge binary decisions using an operational threshold calibrated independently of the configuration under evaluation.

To emulate threshold determination by $\mathcal{V}$, the ten models in $\mathcal{M}_{\text{eval}}$ are divided into calibration and held-out model sets. For each calibration setting, three models are selected as calibration models held by $\mathcal{V}$, while the remaining seven models are reserved for held-out evaluation. All $\binom{10}{3} = 120$ possible selections of the three calibration models are evaluated. For each selection, the same-model directed configurations induced by the three calibration models form the calibration subset of $\Pi_1^{\mathcal{O}}$, whereas the cross-model directed configurations among those models form the calibration subset of $\Pi_0^{\mathcal{O}}$. The corresponding configurations induced by the seven remaining models are used for held-out evaluation. Thus, calibration and held-out evaluation use disjoint model subsets while preserving the same population-scoped identity objective, $\mathcal{D}_C$, ResGen, and decision rule.

The 100 available challenge samples are randomly divided

into two disjoint sets of 50 calibration challenges and 50 test challenges. The calibration challenges are used only for threshold determination, whereas the test challenges are used only for held-out finite-challenge evaluation. For threshold calibration, we fix the calibration aggregation size to $N_{\text{cal}} = 10$. For each calibration-model configuration, non-overlapping groups of N_{cal} calibration challenges are formed, and the corresponding aggregate score is

$$\Gamma_{N_{\text{cal}}} = \frac{1}{N_{\text{cal}}} \sum_{i=1}^{N_{\text{cal}}} D_i^{\text{prob}}. \quad (51)$$

The matching and non-matching calibration configurations therefore induce empirical distributions of $\Gamma_{N_{\text{cal}}}$ using only models held by $\mathcal{V}$ and calibration challenges.

$\tau_{\mathcal{O}}$ is selected as

$$\tau_{\mathcal{O}} = \frac{\max \Gamma_{N_{\text{cal}}}^{\text{same}} + \min \Gamma_{N_{\text{cal}}}^{\text{cross}}}{2}, \quad (52)$$

where the maximum same-model score and minimum cross-model score are computed using only the calibration configurations and calibration challenges. If the calibration distributions are separated such that $\max \Gamma_{N_{\text{cal}}}^{\text{same}} < \min \Gamma_{N_{\text{cal}}}^{\text{cross}}$, $\tau_{\mathcal{O}}$ is selected as the midpoint between these two values. Otherwise, $\tau_{\mathcal{O}}$ is selected as the operating point that minimizes the absolute difference between the false-acceptance and false-rejection rates on the calibration data.

Importantly, in this experiment the resulting $\tau_{\mathcal{O}}$ is determined once for each calibration-model selection and is intentionally kept fixed as the number of verification challenges changes.

For held-out evaluation, we consider $N \in \{1, 3, 5, 10\}$. For each evaluated value of N , non-overlapping groups of N test challenges are used to construct one verification trial. With 50 held-out test challenges, this yields 50, 16, 10, and 5 complete groups per directed model configuration for $N = 1, 3, 5$, and 10, respectively. For $N = 3$, the two remaining challenges are not used in a complete group. $\mathcal{V}$ computes

$$\Gamma_N = \frac{1}{N} \sum_{i=1}^N D_i^{\text{prob}}. \quad (53)$$

Because $N_{\text{val}} = N_{\text{min}} = N$ for every trial in this evaluation, the general acceptance rule in Eq. (36) reduces to

$$\text{Accept} = 1 \iff \Gamma_N \leq \tau_{\mathcal{O}}. \quad (54)$$

Accordingly, the same $\tau_{\mathcal{O}}$ is applied to Γ_1 , Γ_3 , Γ_5 , and Γ_{10} within each calibration-model selection.

For statistical reporting, one verification trial is defined as one value of Γ_N obtained from one directed model configuration and one non-overlapping group of N held-out test challenges under one fixed calibration-model selection. Acceptance of a configuration in $\Pi_0^{\mathcal{O}}$ is counted as a false acceptance, whereas rejection of a configuration in $\Pi_1^{\mathcal{O}}$ is counted as a false rejection. Trials constructed from the same directed model configuration are not regarded as independent model-level samples. For each of the 120 calibration-model selections, FAR, FRR, and AUC are computed separately using the corresponding seven held-out models and test challenges.

TABLE VI

HELD-OUT FINITE-CHALLENGE VERIFICATION USING A FIXED OPERATIONAL THRESHOLD CALIBRATED WITH $N_{\text{cal}} = 10$. FOR EACH OF THE 120 CALIBRATION-MODEL SELECTIONS, THE RESULTING $\tau_{\mathcal{O}}$ IS APPLIED UNCHANGED TO ALL EVALUATED VALUES OF N . THE REPORTED FAR AND FRR ARE THE MEDIAN AND RANGE OVER THE 120 SELECTIONS.

N	Error rate across selections median [min, max]	Min. AUC
1	FAR: 0 [0, 0]% FRR: 0 [0, 0]%	1.000
3	FAR: 0 [0, 0]% FRR: 0 [0, 0]%	1.000
5	FAR: 0 [0, 0]% FRR: 0 [0, 0]%	1.000
10	FAR: 0 [0, 0]% FRR: 0 [0, 0]%	1.000

Across the 120 calibration-model selections, the fixed threshold calibrated using $N_{\text{cal}} = 10$ has a mean of 0.4868, a standard deviation of 0.0114, and a range from 0.4653 to 0.5035. This variation reflects the choice of the three calibration models held by $\mathcal{V}$.

Table VI reports the held-out finite-challenge results obtained using these fixed thresholds. FAR, FRR, and AUC are computed separately for each of the 120 calibration-model selections. Because the 120 calibration-model selections overlap, we report the distribution of FAR and FRR across selections. No false acceptance or false rejection was observed for any of the 120 calibration-model selections or for any evaluated value of $N \in \{1, 3, 5, 10\}$. Accordingly, the median, minimum, and maximum of the split-specific FAR and FRR are all zero. The minimum split-specific AUC is also 1.000 for every evaluated value of N . Thus, even for $N = 1$, the fixed threshold calibrated from Γ_{10} on the calibration models held by $\mathcal{V}$ separated all evaluated matching and non-matching held-out trials.

In the present experiment, the observable separation is sufficiently large that perfect held-out classification is retained for all evaluated values of N . However the 120 calibration-model selections should not be interpreted as 120 independent experimental replications. Because the selections overlap, each same-model directed configuration appears in the held-out set for $\binom{9}{3} = 84$ selections, whereas each cross-model directed configuration appears in the held-out set for $\binom{8}{3} = 56$ selections. The same challenge instances are also reused across different calibration-model selections. The split-specific results therefore constitute an exhaustive sensitivity analysis with respect to the choice of calibration models rather than independent estimates from randomly repeated experiments.

At the distinct-configuration level, the held-out evaluations collectively cover all 10 elements of $\Pi_1^{\mathcal{O}}$ and all 90 elements of $\Pi_0^{\mathcal{O}}$. None of the 90 non-matching configurations was accepted under any calibration-model selection in which that configuration was held out. Similarly, none of the 10 matching configurations was rejected. Thus, the observed zero-error result is shared by all distinct configurations in the instantiated operating populations and is not produced solely by pooling repeated challenge-level observations.

These results indicate that, within the instantiated operating

world, the observable separation characterized in Fig. 5 can be converted into finite-challenge decisions using a fixed operational threshold estimated from separate models held by $\mathcal{V}$. Accordingly, when $\mathcal{V}$ observes an accepted verification result for a suspicious configuration, the demonstrated challenge-solving performance is interpreted as being more consistent with $\Pi_1^{\mathcal{O}}$ than with $\Pi_0^{\mathcal{O}}$ within the calibrated scope. Together with freshness, network isolation, and the model-based response-generation assumption, this provides an empirical basis for inferring that $\mathcal{P}$ locally possesses a model satisfying the declared identity relative to M^{cloud} .

The absence of false rejection provides empirical evidence of finite-challenge completeness for $\Pi_1^{\mathcal{O}}$ under the evaluated response-generation setting, whereas the absence of false acceptance provides empirical evidence of soundness-related behavior for $\Pi_0^{\mathcal{O}}$ under the same condition. The reported empirical FAR and FRR summarize the observed finite-challenge error behavior of the evaluated configurations under the instantiated $\mathcal{D}_C$, ResGen, and deployment pipeline. They should not be interpreted as population-level guarantees for model configurations or response-generation algorithms outside this experimental scope. The model configurations also share constituent models and should not be regarded as independent draws from a broader population of AI models. Accordingly, the results do not provide a universal threshold guarantee, a broader population-level confidence interval, or proof that every non-matching model must fail the selected challenge family.

4) *Assessment of the Protocol Requirements:* The evaluated CNN instantiation satisfies the abstract requirements within the declared experimental scope as follows.

a) *Completeness:* For $\Pi_1^{\mathcal{O}}$, $M = M^{\text{cloud}}$ and ResGen = I-FMPC. Table VI reports no held-out rejection for any evaluated N . These observations provide empirical evidence of finite-challenge completeness for $\Pi_1^{\mathcal{O}}$.

b) *Soundness:* For $\Pi_0^{\mathcal{O}}$, the same ResGen searches using S_M while the requirement is evaluated using M^{cloud} . Table VI reports no false acceptance for any evaluated N . This soundness-related evidence applies only to the 90 declared non-matching configurations and the fixed ResGen used in the experiment.

c) *Property-Demanding Challenge Design:* Each C_i varies $\mathbf{x}_i$, c'_i , and $p_{c'_i, t}^{\text{target}}$, and each valid W_i must remain within ε_{val} of $\mathbf{x}_i$. Fig. 5 shows that, for configurations in $\Pi_1^{\mathcal{O}}$, the resulting witnesses consistently satisfy the specified fine-grained probability requirements on M^{cloud} , whereas for configurations in $\Pi_0^{\mathcal{O}}$, the witnesses generally fail to transfer the same requirements to M^{cloud} .

d) *Witness Validity:* All reported witnesses satisfy $\varepsilon_{\text{gen}} \leq \varepsilon_{\text{val}}$, yielding $N_{\text{val}} = N$ in every reported trial. With $N_{\text{min}} = N$, acceptance therefore cannot be obtained by returning valid witnesses only for a favorable subset of challenges.

e) *Freshness and Network Isolation:* Base inputs are not reused within a session and the remaining challenge components are freshly sampled from $\mathcal{D}_C$. The operational interpretation additionally assumes the network-isolation condition of Section III. Together with the calibrated matching

decision, these conditions satisfy the freshness and network-isolation requirement within the evaluated model-based scope.

V. CONCLUDING REMARKS

This paper proposed TP-CRIV, a challenge-response framework for third-party identity verification of AI models deployed through remote services. TP-CRIV considers a setting in which the verifier $\mathcal{V}$ has neither white-box nor API access to the claimant's model M , interacts with the suspicious deployed model M^{cloud} only through its ordinary black-box service interface, and requires no protocol-specific cooperation from the service provider. Under these conditions, TP-CRIV enables $\mathcal{V}$ to obtain empirical evidence as to whether the prover $\mathcal{P}$ locally possesses a model satisfying the identity specified by the intended verification objective $\mathcal{O}$.

For each $\mathcal{O}$, matching and non-matching operating situations, $\Pi_1^{\mathcal{O}}$ and $\Pi_0^{\mathcal{O}}$, are specified a priori. A model-dependent property S is then used to construct fresh property-demanding challenges, and the resulting challenge-solving performance is interpreted using calibration on independent reference configurations. Within the model-based response-generation scope, freshness and network isolation exclude online external assistance after challenge disclosure. Matching-consistent performance therefore provides an empirical basis for inferring that the prover $\mathcal{P}$ possesses a model satisfying the identity declared by $\mathcal{O}$. This inference is statistical and does not establish training provenance or legal ownership.

We instantiated TP-CRIV using ten ImageNet-pretrained CNN models with ResGen = I-FMPC. Within the evaluated operating population $\Pi^{\mathcal{O}}$, same-model configurations were assigned to $\Pi_1^{\mathcal{O}}$, whereas cross-model configurations were assigned to $\Pi_0^{\mathcal{O}}$. The two situations exhibited clear score separation, and independently calibrated thresholds produced no observed false acceptance or false rejection for $N \in \{1, 3, 5, 10\}$ across all 120 calibration-model selections. These results demonstrate feasibility within the explicitly evaluated operating scope rather than universal soundness over arbitrary models or response-generation mechanisms.

Several directions remain for extending the framework and its operational guarantees.

- **Broader operating populations:** Future implementations should extend both $\Pi_1^{\mathcal{O}}$ and $\Pi_0^{\mathcal{O}}$ according to the intended verification objective $\mathcal{O}$. These populations may include independently trained, fine-tuned, pruned, quantized, extracted, distilled, and otherwise transformed models, with each configuration assigned to the matching or non-matching situation according to the identity semantics of $\mathcal{O}$. For example, a transformed model may belong to $\Pi_0^{\mathcal{O}}$ under an exact-instance objective but to $\Pi_1^{\mathcal{O}}$ under a lineage-oriented objective. Broader populations would enable more reliable characterization of FAR, FRR, threshold transfer, and soundness within the declared operating scope.
- **Broader non-matching prover strategies:** Future work should consider a broader range of response-generation strategies ResGen available to provers whose model configurations belong to $\Pi_0^{\mathcal{O}}$. Even for the

same non-matching configuration, stronger optimization, ensemble-based strategies, or other model-based procedures may change challenge-solving performance and false-acceptance behavior. A further extension is to consider non-model response mechanisms, such as learned challenge-to-witness generators, which lie outside the present model-based response-generation and soundness scope.

- **Challenge design and deployment conditions:** Future work should investigate challenge-selection strategies, the relationship between challenge diversity and the number of challenges N , practical enforcement or auditing of network isolation, and robustness to service-side preprocessing, model modification, and stochastic inference.
- **Broader AI-model instantiations:** Future work should instantiate and evaluate TP-CRIV for a wider range of AI model types, including generative, language, multimodal, and other AI systems. For each model type, suitable model-dependent properties S_M , property-demanding challenge families, and response-generation algorithms ResGen should be identified and evaluated according to the intended verification objective $\mathcal{O}$ and operating scope.

REFERENCES

- [1] Y. Li, H. Wang, and M. Barni, "A survey of deep neural network watermarking techniques," *Neurocomputing*, vol. 461, pp. 171–193, 2021.
- [2] Y. Sun, T. Liu, P. Hu, Q. Liao, S. Fu, N. Yu, D. Guo, Y. Liu, and L. Liu, "Deep intellectual property protection: A survey," *arXiv preprint arXiv:2304.14613*, 2023.
- [3] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *Proc. 25th USENIX Security Symposium*, pp. 601–618, 2016.
- [4] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4949–4958, 2019.
- [5] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proc. ACM International Conference on Multimedia Retrieval*, pp. 269–277, 2017.
- [6] H. Chen, B. Darvish Rouhani, C. Fu, J. Zhao, and F. Koushanfar, "Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models," in *Proceedings of the 2019 International Conference on Multimedia Retrieval*, pp. 105–113, ACM, 2019.
- [7] Y. Adi, C. Baum, M. Cissé, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *Proc. 27th USENIX Security Symposium*, pp. 1615–1631, 2018.
- [8] B. D. Rouhani, H. Chen, and F. Koushanfar, "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 (ASPLOS)*, pp. 485–497, ACM, 2019.
- [9] E. Le Merrer, P. Pérez, and G. Trédan, "Adversarial frontier stitching for remote neural network watermarking," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9233–9244, 2020.
- [10] P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon, "The stable signature: Rooting watermarks in latent diffusion models," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22409–22420, 2023.
- [11] H. Liu, W. Zhang, B. Li, B. Ghanem, and J. Schmidhuber, "Lazy layers to make fine-tuned diffusion models more traceable," *arXiv preprint arXiv:2405.00466*, 2024.
- [12] R. Zhang, S. S. Hussain, P. Neekhara, and F. Koushanfar, "REMARK-LLM: A robust and efficient watermarking framework for generative large language models," in *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 1813–1830, USENIX Association, 2024.

- [13] X. Cao, J. Jia, and N. Z. Gong, "IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary," in *Proc. ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, pp. 14–25, 2021.
- [14] J. Zhao, Q. Hu, G. Liu, X. Ma, F. Chen, and M. M. Hassan, "Afa: Adversarial fingerprinting authentication for deep neural networks," *Comput. Commun.*, vol. 150, p. 488–497, Jan. 2020.
- [15] N. Lukas, Y. Zhang, and F. Kerschbaum, "Deep neural network fingerprinting by conferrable adversarial examples," in *9th International Conference on Learning Representations, ICLR 2021*, OpenReview.net, 2021.
- [16] S. Wang and C.-H. Chang, "Fingerprinting deep neural networks – a deepfool approach," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2021.
- [17] Q. Li and G. Xu, "Deep neural network fingerprinting by general examples," *Neurocomputing*, vol. 665, p. 132184, 2026.
- [18] Z. Peng, S. Li, G. Chen, C. Zhang, H. Zhu, and M. Xue, "Fingerprinting deep neural networks globally via universal adversarial perturbations," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13420–13429, 2022.
- [19] M. Gubri, D. Ulmer, H. Lee, S. Yun, and S. J. Oh, "TRAP: Targeted random adversarial prompt honeypot for black-box identification," in *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11496–11517, Association for Computational Linguistics, 2024.
- [20] A. Godinot, E. L. Merrer, C. Penzo, F. Taïani, and G. Trédan, "Queries, representation & detection: The next 100 model fingerprinting schemes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 16817–16825, 2025.
- [21] Y. Li, Z. Zhang, B. Liu, Z. Yang, and Y. Liu, "ModelDiff: Testing-based DNN similarity comparison for model reuse detection," in *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA)*, pp. 139–151, ACM, 2021.
- [22] J. Chen, J. Wang, T. Peng, Y. Sun, P. Cheng, S. Ji, X. Ma, B. Li, and D. Song, "Copy, right? a testing framework for copyright protection of deep learning models," in *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 824–841, IEEE, 2022.
- [23] R. Yu and X. Wang, "Neural lineage," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4797–4807, 2024.
- [24] I. Nikolic, T. Baluta, and P. Saxena, "Model provenance testing for large language models," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [25] Z. Ghodsi, T. Gu, and S. Garg, "SafetyNets: Verifiable execution of deep neural networks on an untrusted cloud," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4675–4684, 2017.
- [26] C. Weng, K. Yang, X. Xie, J. Katz, and X. Wang, "Mystique: Efficient conversions for Zero-Knowledge proofs with applications to machine learning," in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 501–518, USENIX Association, Aug. 2021.
- [27] T. Liu, X. Xie, and Y. Zhang, "zkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2968–2985, Association for Computing Machinery, 2021.
- [28] S. Lee, H. Ko, J. Kim, and H. Oh, "vCNN: Verifiable convolutional neural network based on zk-SNARKs," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 4254–4270, 2024.
- [29] B.-J. Chen, S. Waiwitlikhit, I. Stoica, and D. Kang, "ZKML: An optimizing system for ML inference in zero-knowledge proofs," in *Proceedings of the Nineteenth European Conference on Computer Systems*, pp. 560–574, Association for Computing Machinery, 2024.
- [30] W. Qu, Y. Sun, X. Liu, T. Lu, Y. Guo, K. Chen, and J. Zhang, "zkGPT: An efficient non-interactive zero-knowledge proof framework for LLM inference," in *34th USENIX Security Symposium (USENIX Security 25)*, (Seattle, WA), pp. 2045–2063, USENIX Association, 2025.
- [31] T. Sano, M. Kuribayashi, M. Sakai, S. Isobe, and E. Koizumi, "Ownership verification of dnn models using white-box adversarial attacks with specified probability manipulation," in *Proc. European Signal Processing Conference (EUSIPCO)*, pp. 795–799, 2025.
- [32] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
- [35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., 2019.
- [36] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, vol. 2, pp. 3320–3328, 2014.
- [37] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, vol. 1, pp. 1135–1143, 2015.
- [38] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2704–2713, 2018.
- [39] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.
- [40] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2015.
- [41] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2017.
- [42] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.
- [43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [44] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.