FBI: Fingerprinting models with Benign Inputs

Thibault Maho, Teddy Furon, Erwan Le Merrer

Centre Inria de l'Université de Rennes, France

August 8, 2022

Abstract

Recent advances in the fingerprinting of deep neural networks detect instances of models, placed in a black-box interaction scheme. Inputs used by the fingerprinting protocols are specifically crafted for each precise model to be checked for. While efficient in such a scenario, this nevertheless results in a lack of guarantee after a mere modification (*e.g.* retraining, quantization) of a model.

This paper tackles the challenges to propose i) fingerprinting schemes that are resilient to significant modifications of the models, by generalizing to the notion of model families and their variants, ii) an extension of the fingerprinting task encompassing scenarios where one wants to fingerprint not only a precise model (previously referred to as a *detection* task) but also to identify which model family is in the black-box (*identification* task).

We achieve both goals by demonstrating that benign inputs, that are unmodified images, for instance, are sufficient material for both tasks. We leverage an information-theoretic scheme for the identification task. We devise a greedy discrimination algorithm for the detection task. Both approaches are experimentally validated over an unprecedented set of more than 1,000 networks.

1 Introduction

Fingerprinting classifiers aim at deriving a signature uniquely identifying a model, like the human fingerprint's minutiae in biometry. This is essentially a black-box problem: the classifier to be identified is in a black-box in the sense that one can just make some queries and observe its outputs. For instance, the model is embedded in a chip or we can query it through an API (MLaaS).

The main application that related works [1, 2, 3, 4, 5] target is the proof of ownership. An accurate deep neural network is a valuable industrial asset due to the know-how for training it, the difficulty of gathering a well-annotated training dataset, and the required computational resources to learn its parameters. The cost of GPT-3, one of the biggest and most accurate NLP models is estimated to 4.6 million dollars[1]. In this context, the entity identifying a black-box wants to *detect* whether it is not a stolen model of her.

Another at least as critical application is information gain. For instance, an attacker willing to delude the classifier first gains some knowledge about the remote model, or a company wants to determine which model is in use in a competitor's production system. This application has been left aside as of today, and we tackle it under the notion of the fingerprinting *identification* task.

For clarity, we name Alice the entity willing to identify the model that Bob has embedded in the black-box.

Challenges The biggest difficulty is that there exist plenty of ways to modify a model while maintaining its intrinsic good accuracy. These procedures simplify a network (quantification of the weights and/or activations, pruning, see *e.g.* [6]), or make it more robust (preprocessing of the input, adversarial re-training [7]). We hereafter name a modified model a *variant*. These mechanisms were not a priori designed to make fingerprinting harder but they leave room for Bob to tamper with the fingerprint of a model. We assume that Alice also knows some of these procedures. Yet, they are often defined by many parameters and among them scalars so that there is virtually an infinite continuum of variants. Like in biometry, the fingerprint should be discriminative enough for being unique per model but also sufficiently robust to identify a variant.

The approaches in the literature have two noticeable common pillars. They use the frontiers in the input space drawn by a classifier as the fingerprint, *i.e.* the unique signature identifying the model [2, 4, 5]. Two neural networks sharing the same architecture, the same training set, and the same training procedure are different because the training is stochastic (like the Stochastic Gradient Descent). This makes their frontiers in the input space not overlap. Most of the papers in the literature are looking for discriminative deviations of these frontiers. Second, the key task is detection: Alice makes a guess about the model in the black-box and then she queries these specific queries to test whether her hypothesis holds [2, 8, 4, 5].

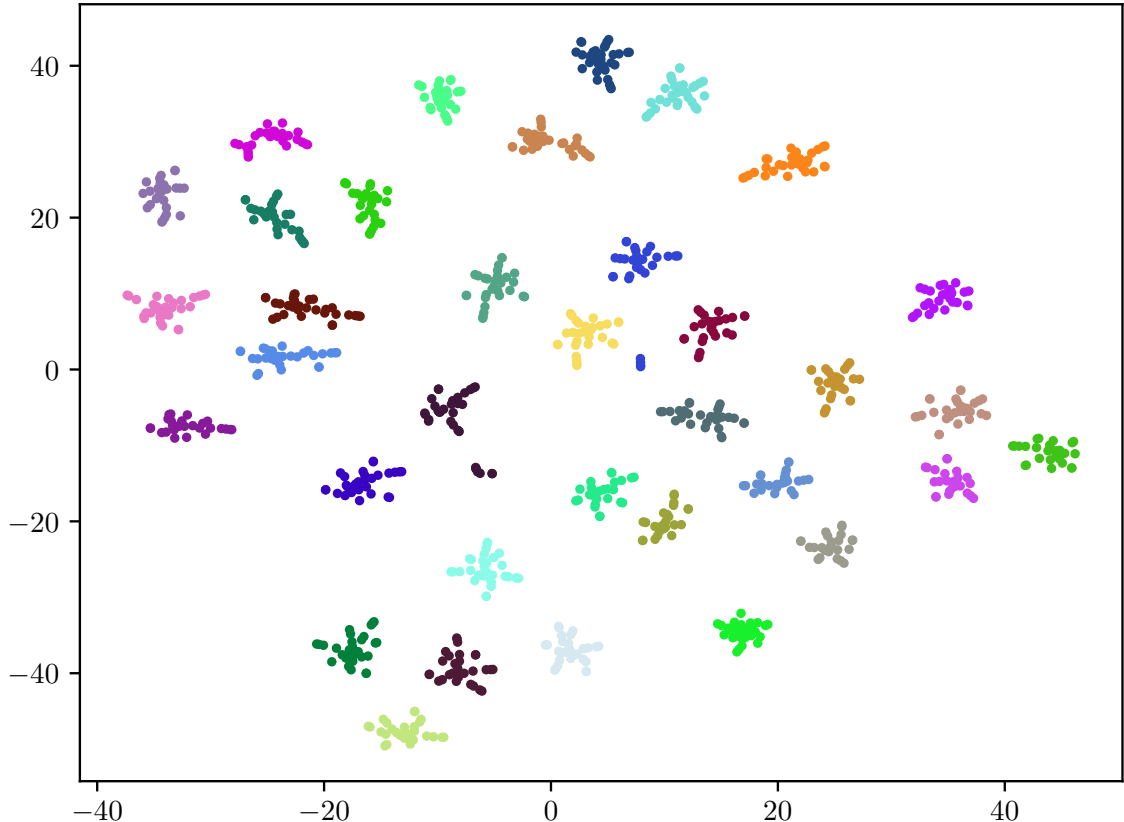[1]https://lambdalabs.com/blog/demystifying-gpt-3/

Figure 1: A t-SNE representation of the pairwise distances of 1081 different models: 10 types of variation applied on 35 off-the-shelves vanilla models for ImageNet with different parameters (listed in App. B.2). This work exploits the clear separability (clusters of consistent colors) observed in the decisions of these models. Confusions yet happen (model colors further apart from their cluster), but are under scrutiny for the tracking of false positive identification.

Rationale   We find that before resorting to the crafting of inputs for fingerprinting models, the use of benign inputs has not been thoroughly investigated. The use of benign inputs would constitute a certain advantage, as it would remove the need for the design of often complex ways to efficiently craft these. It is less prone to defenses being implemented on Bob's side (*e.g.* rejection based on the distance to the decision frontier [9]). A second salient observation is the restriction of previous works on the detection task. The more general possibility to identify a model or a family of models inside the black-box remains unstudied. Our work thus differs from related works on these two key aspects: *i)* we do not forge any specific input but use regular benign inputs (so that we do not need to probe the input space to discover the decision frontiers), and *ii)* we directly identify models using their intrinsic classification difficulties on a handful of inputs, this is more efficient than sequentially detecting models until finding the good one.

In a nutshell, when Bob has picked a model among a set of networks known by Alice, then our solution is essentially deterministic: Alice has to find the sequence of inputs of minimum length to identify the black-box. We apply a greedy algorithm that carefully selects the input to iteratively narrow down the set of suspects, *i.e.* candidate models, until it becomes a singleton. Approximation theory tells that this is suboptimal but we report that in practice many networks are indeed identified within less than three queries.

When an attack by Bob has made a variant of a model, the output of it may not match the output of any known model. We then use C.E. Shannon's information theory to measure the statistical similarity between the outputs of two models. As an appetizer, Figure 1 depicts the t-SNE representation from the pairwise distances within a set of 35 vanilla models and their variants. The model families are well clustered in the sense that variants are closer to their original network than any other model. Alice may not identify precisely the variant of the model but at least she can identify its family, *i.e.* infer which was the original vanilla model and even which kind of variation was applied to it.

Contributions   Our contribution is fourfold.
1) We demonstrate that the mere use of benign images is enough to accomplish high success rates for finger-

printing modern models. This is to be opposed to the computationally demanding task of crafting inputs for that same goal.

2) The fingerprinting detection task, introduced by state-of-the-art works, is complemented with the introduction of the identification task. We frame the latter as an information theoretical problem.

3) We present a distance based on the empirical Mutual Information, gauging how close two models are. This distance permits to generalize the notion of attacks on models through the notion of model families and variants.

4) We perform extensive experimentation by considering more than 1,000 classification models on ImageNet. A head-to-head comparison with the two related works reports significant improvements w.r.t. accuracy in the detection task.

Document structure Section 2 is a threat analysis listing all the working assumptions in our work. The next two sections deal both with detection (Alice verifies her hypothesis about the black-box) and identification (Alice discovers which model is in the black-box) but under two scenarios: Section 3 builds on the fact that Bob has picked a model among the set known by Alice, whereas Section 4 assumes that the black-box may be an unknown model. Both sections contain experimental results. Section 6 is devoted to the related works and the benchmark with state-of-the-art detection schemes. The appendices contain a summary of the notations, details about the experimental protocol, and supplementary results.

2 Threat Model

This section details the goals of Alice and Bob.

2.1 Bob: keeping his model anonymous

2.1.1 Goals

Bob is playing first by secretly selecting a model and putting it in the black-box. This model can be a vanilla model or a variant of a known model. A variant is created by applying on a given vanilla model m the procedure V parametrized by $\theta \in \Theta$ which describes the type of modification and the associated parameters; this can be thought of as an attack by Bob on the vanilla model to harden identification. We denote such a variant by $\mathsf{v} = \mathsf{V}(\mathsf{m}, \theta)$.

The goal of Bob is to offer an accurate black-box classifier while maintaining the 'anonymity' of the model in use. The first requirement is that a small loss in performance is tolerated. If a variant does not comply with this criterion then Bob cannot consider it as an option. In classification, the performance of a model m is often gauged by the top-1 accuracy, denoted $\mathsf{acc}(\mathsf{m})$. We formalize this requirement as

$$\frac{\mathsf{acc}(\mathsf{m}) - \mathsf{acc}(\mathsf{V}(\mathsf{m}, \theta))}{\mathsf{acc}(\mathsf{m})} < \eta, \tag{1}$$

where $\eta > 0$ is the tolerance (set to 15% in our experimental work).

2.1.2 Ressources

The second requirement is more delicate. We first need to limit the power of Bob. If Bob creates an accurate model *ex nihilo*, then Alice can pursue neither detection nor identification. We assume that Bob cannot train such a model from scratch because he lacks good training data, expertise in machine learning, or computing resources. This also means that Bob cannot retrain a model, or only up to a limited extent. In other words, the complexity of the procedure creating $\mathsf{v} = \mathsf{V}(\mathsf{m}, \theta)$ ought to be much smaller than the effort spent at training the original model m.

Our experimental work considers two kinds of procedures:

1) modification of the input: $\mathsf{v}(x) = \mathsf{m}(\mathsf{T}(x, \theta))$. Classifiers are robust to benign transformations of the input. As far as images are concerned, the transformation T can be JPEG compression, histogram equalization, blurring, *etc.* In the same spirit, Randomized Smoothing consists in adding noise patterns to the input and aggregating the predicted classes into one single output.

2) modification of the model: $\mathsf{v}(x) = \mathsf{T}(\mathsf{m}, \theta)(x)$: The transform T slightly changes the model weights by quantization, pruning, adversarial retraining, finetuning... Some of these procedures require small retraining with few resources so as not to lose too much accuracy.

In the sequel, the model in the black-box is denoted by b and $\mathcal{B}$ is the set of possibilities: $\mathsf{b} \in \mathcal{B}$. This set is defined as:

$$\mathcal{B} := \{\mathsf{v} = \mathsf{V}(\mathsf{m}, \theta) : \mathsf{m} \in \mathcal{P}, \theta \in \Theta, \mathsf{acc}(\mathsf{v}) > (1 - \eta)\mathsf{acc}(\mathsf{m})\}, \tag{2}$$

where $\mathcal{P}$ is a set of vanilla models and Θ a set of transformations (encompassing the identity $\mathsf{v} = \mathsf{m}$).

2.2 Alice: disclosing the remote model

2.2.1 Goals

The task of Alice is to disclose which model is in the black-box. This has two flavours: detection or identification.

Detection (denoted by D) means that Alice performs a hypothesis test. She first makes a hypothesis about the black-box, then makes some queries, and finally decides whether the hypothesis holds based on the outputs of the black-box. The outcome of the detection is thus binary: Alice's hypothesis is deemed correct or not. This is the nominal use case in the related works [1, 2, 3, 4, 5].

Identification (denoted by I) means that Alice has no prior about the model in the black-box. She makes queries and processes the outputs to finally make a guess. The outcome is either the name of a model she knows, or the absence of a decision if she has not enough evidence.

2.2.2 Knowledge about the black-box

The second crucial point is her knowledge about the black-box. Alice can only detect or identify a relation to a model she knows: this means that she has an implementation of this model, which she can freely test in white-box access, typically locally on her machine. We denote the set of models known by Alice by $\mathcal{A}$.

As by the very definition of a *variant*, Alice may know some of them but not all of them. For instance, some procedures V admit a real number as a parameter. Therefore, there is virtually an infinite number of variants. This leads to the convenient notion of a model *family*, we now introduce under three flavours:

- $\mathcal{F}(\mathsf{m})$: This family is the set of all variants made from the original vanilla model m:

$$\mathcal{F}(\mathsf{m}) \coloneqq \{v = \mathsf{V}(\mathsf{m}, \theta) : \theta \in \Theta\}. \tag{3}$$

- $\mathcal{F}(\mathsf{m}, \Psi)$: This family is the set of all variants made from the original vanilla model m by a specific procedure:

$$\mathcal{F}(\mathsf{m}, \Psi) \coloneqq \{v = \mathsf{V}(\mathsf{m}, \theta) : \theta \in \Psi \subset \Theta\}, \tag{4}$$

  where Ψ denotes the subset of parameters related to this specific procedure.

- $\mathcal{F}(\mathsf{m}, \{\theta\})$: This family is a singleton composed of a particular variant:

$$\mathcal{F}(\mathsf{m}, \{\theta\}) \coloneqq \{v = \mathsf{V}(\mathsf{m}, \theta)\}. \tag{5}$$

With these definitions in mind, detection is based on the hypothesis that the black-box belongs to a given family, while Identification looks for the family the black-box belongs to.

2.2.3 Resources

A third element is the resources of Alice. We already mention the set $\mathcal{A}$ containing some vanilla models and few variants of theirs. She also has a collection of typical inputs, *i.e.* a testing dataset. We suppose that they are statistically independent and distributed as the data in the training set of the models. In the sequel, the collection of inputs is denoted $\mathcal{X} = \{x_1, \ldots, x_N\}$.

In the end, be it for detection or identification, Alice selects some elements of $\mathcal{X}$ for querying the black-box. We denote this by an ordered list of indices: $q_{1:\ell} = (q_1, \ldots, q_\ell) \in [\![N]\!]^\ell$, where $[\![N]\!] \coloneqq \{1, \ldots, N\}$. This means that Alice first queries x_{q_1}, and then x_{q_2} and so forth. The outputs of the black-box are denoted as $z_{1:\ell} = (z_{q_1}, \ldots, z_{q_\ell})$, with $z_{q_i} = \mathsf{b}(x_{q_i})$.

2.3 The classifier in the black-box

The black-box works as any classifier. We denote the set of possible classes $\mathcal{C}$. The output $z = \mathsf{b}(x)$ for input x is the first k classes ordered by their predicted probabilities (*i.e.* the top-k). It means that z is an ordered list in $\mathcal{C}^k$: $z = (c_1, \ldots, c_k)$. The set $\mathcal{Z}_k$ of possible outcomes has a size as big as $(C)_k \coloneqq C(C-1) \ldots (C-k+1)$. The black-box only discloses these classes, not the associated predicted probabilities. In the experimental work, the size of $\mathcal{C}$ is 1,000 (ImageNet) and $k \in \{1, 3, 5\}$ which is usual in several image classification APIs. We assume that the considered models and variants have an accuracy which is not perfect. The top-1 accuracy of ImageNet classic models ranges from 70% to 85%.

2.4 Summary

This paper considers scenarios which are labeled as $(\mathsf{Task}, \mathcal{F}, \mathcal{A}, k)$ where $\mathsf{Task} \in \{\mathsf{D}, \mathsf{I}\}$ (Detection or Identification), $\mathcal{F}$ is the kind of family that will be inferred by Alice, $\mathcal{A}$ is the set of models known by Alice, and k indicates that the output of the black-box is the top-k classes. There is a clear cut between the following two cases:

1) Walled garden: $\mathcal{A} = \mathcal{B}$. We impose that the black-box is one of the networks known by Alice.
2) Open world: $\mathcal{A} \subsetneq \mathcal{B}$. The black-box may not be a model known by Alice. This is the case when Bob uses an unknown variant, for instance.

   This distinction drives the structure of the next sections because our solutions are of different nature.

3 The black-box is a known model

Under the assumption that $\mathcal{A} = \mathcal{B}$, Alice achieves her goal when she correctly guesses which family the black-box belongs to. The alternative is to fail to gather enough evidence to make a decision. For the sake of clarity, we explain our procedure for a given family $\mathcal{F} \subset \mathcal{A}$, which can be one of the three types of families presented in Sect. 2.2.2.

Alice has a set of models composed of some vanilla models $\mathcal{P} = \{\mathsf{m}_1, \ldots, \mathsf{m}_M\}$ and some variants of theirs. Alice also has the collection of benign inputs $\mathcal{X} = (x_1, \ldots, x_N)$. Offline, she creates a database of $|\mathcal{A}|N$ outputs $(\mathsf{m}(x_j))_{\mathsf{m} \in \mathcal{A}, j \in [\![N]\!]}$.

Let $\mathcal{D}$ be a subset of $\mathcal{A}$. We define by $\mathcal{D}(x) := \{\mathsf{m}(x) | \mathsf{m} \in \mathcal{D}\}$ the set of labels predicted by the models in $\mathcal{D}$ for input x. With abuse of notations, $\mathcal{D}(x_{q_{1:\ell}})$ is the set of the concatenation of labels predicted by the models in $\mathcal{D}$ for the entries $(x_{q_1}, \ldots, x_{q_\ell})$. Conversely, $\mathcal{M}(x, y, \mathcal{D}) := \{\mathsf{m} | \mathsf{m} \in \mathcal{D}, \mathsf{m}(x) = y\}$ lists the models in $\mathcal{D}$ predicting y when x is submitted.

3.1 Detection $(\mathsf{D}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$

Alice first makes a hypothesis about a family $\mathcal{F}$, and her goal is to discover whether the outcome is *positive* ($\mathsf{b} \in \mathcal{F}$) or *negative* ($\mathsf{b} \in \mathcal{A} \backslash \mathcal{F}$). We assume that Alice is convinced about her hypothesis and that she hopes for a *positive*. Our procedure thus focuses on reducing the number of models in $\mathcal{A} \backslash \mathcal{F}$ likely to be the black-box.

Alice uses a greedy algorithm which leverages the information about the black-box retrieved from the previous queries. According to the outputs of the black-box, several models can be discarded. At step ℓ, $(\mathcal{A} \backslash \mathcal{F})^{(\ell)}$ (resp. $\mathcal{F}^{(\ell)}$) denote the subset of models in $\mathcal{A} \backslash \mathcal{F}$ (resp. $\mathcal{F}^{(\ell)}$) which agree with the previous outputs. These are candidates in the sense that they could be the black-box model.

$$(\mathcal{A} \backslash \mathcal{F})^{(\ell)} := \bigcap_{i=1}^{\ell} \mathcal{M}(x_{q_i}, \mathsf{b}(x_{q_i}), \mathcal{A} \backslash \mathcal{F}). \tag{6}$$

Initially, all the models are candidates: $(\mathcal{A} \backslash \mathcal{F})^{(0)} = \mathcal{A} \backslash \mathcal{F}$ and $\mathcal{F}^{(0)} = \mathcal{F}$. At step $\ell + 1$, the greedy algorithm sorts the inputs that have not yet been queried according to a score. This score $s^{(\ell+1)}(x)$ reflects how much the set of candidates $(\mathcal{A} \backslash \mathcal{F})^{(\ell)}$ reduces if input x is submitted next. It is thus based on the quantities $|\mathcal{M}(x, y, (\mathcal{A} \backslash \mathcal{F})^{(\ell)})|$.

Expectation Convinced that her hypothesis is correct, Alice computes the expectation of the number of candidate models outside the family after querying input x. The average is weighted by the number of models in $\mathcal{F}^{(\ell)}$ predicting a particular label, as if the black-box were randomly picked in $\mathcal{F}^{(\ell)}$:

$$s^{(\ell+1)}(x) = \sum_{y \in \mathcal{F}^{(\ell)}(x)} \left| \mathcal{M}(x, y, (\mathcal{A} \backslash \mathcal{F})^{(\ell)}) \right| \frac{|\mathcal{M}(x, y, \mathcal{F}^{(\ell)})|}{|\mathcal{F}^{(\ell)}|} \tag{7}$$

Worst Case Instead of a sum, Alice takes the maximum which corresponds to the worst case, at the same time likely to happen and reducing the list of candidates by a small amount if any:

$$s^{(\ell+1)}(x) = \max_{y \in \mathcal{F}^{(\ell)}(x)} \left| \mathcal{M}(x, y, (\mathcal{A} \backslash \mathcal{F})^{(\ell)}) \right| \frac{|\mathcal{M}(x, y, \mathcal{F}^{(\ell)})|}{|\mathcal{F}^{(\ell)}|}. \tag{8}$$

Alice then submits one of the inputs with the lowest score:

$$q_{\ell+1} \in \arg\min_k s^{(\ell+1)}(x_k). \tag{9}$$

Our procedure stops after L iterations when meeting one of the three stopping criteria:

- $(\mathcal{A}\backslash\mathcal{F})^{(L)} = \emptyset$: The detection result is *positive*. No model outside the family responds like the black-box. The black-box is part of the family since we assume it belongs to $\mathcal{A}$.

- $\mathcal{F}^{(L)} = \emptyset$: The detection result is *negative*. The responses of the black-box are different from the ones of the models in the family $\mathcal{F}$.

- $\min_k s^{(L+1)}(x_k) = |(\mathcal{A}\backslash\mathcal{F})^{(L)}|$: The detection failed. All the remaining models in $(\mathcal{A}\backslash\mathcal{F})^{(L)}$ and in $\mathcal{F}^{(L)}$ produce the same prediction no matter which input is submitted. It is therefore impossible to discern them.

3.2 Identification $(\mathsf{I}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$

Identification means that Alice makes a partition of her set of models in disjoint families: $\mathcal{A} = \cup_{i=1}^{n_\mathcal{F}} \mathcal{F}_i$. Her goal is to identify which family the black-box belongs to.

It is easy to base a verification procedure onto a detection scheme assuming there is no failure. Alice arbitrarily orders the families, and sequentially tests the hypotheses until she finds a match. The expected number of queries is given by (see the proof in App. C):

$$\mathbb{E}(L) = \frac{1}{n_\mathcal{F}} \sum_{j=1}^{n_\mathcal{F}} \mathbb{E}\left(L_j^{\mathsf{pos}}\right) + \frac{n_\mathcal{F} - 1}{2n_\mathcal{F}} \sum_{j=1}^{n_\mathcal{F}} \mathbb{E}\left(L_j^{\mathsf{neg}}\right) \tag{10}$$

where $(\mathbb{E}(L_j^{\mathsf{pos}}), \mathbb{E}(L_j^{\mathsf{neg}}))$ are the expected number of queries necessary for taking a positive or negative decision about the *detection* of the hypothesis $\mathcal{F}_j$.

We propose a better approach based on a greedy algorithm similar to the detection one. Suppose that Alice has already submitted ℓ queries to the black-box. By comparing the outputs of the black-box and of the models she knows, she is able to distinguish models which are not in the black-box from models likely to be in the black-box. This list of remaining models is denoted $\mathcal{A}^{(\ell)}$. The goal of Alice is to reduce the set of candidates to a single family $\mathcal{F}_i$, not knowing in advance the model Bob placed in the black-box:

$$\exists i, \mathcal{F}_i \subset \mathcal{A}^{(\ell)}. \tag{11}$$

In the beginning, all the models are possibly in the black-box, *i.e.* $\mathcal{A}^{(0)} = \mathcal{A}$. At step $\ell + 1$, the greedy algorithm chooses the best input to query next knowing $\mathcal{A}^{(\ell)}$. Alice may resort to the following heuristics.

Expectation. Alice supposes that the black-box is randomly chosen uniformly in the set of remaining models $\mathcal{A}^{(\ell)}$. For any input x not queried yet, she computes the expectation of the numbers of remaining families if that input were selected next, *i.e.* $\{\mathcal{F}_i | \mathcal{F}_i \cap \mathcal{A}^{(\ell+1)} \neq \emptyset\}$. She randomly chooses among the inputs which minimize this figure:

$$s^{(\ell+1)}(x) = \sum_{y \in \mathcal{A}^{(\ell)}(x)} \left[\sum_{i=1}^{n_\mathcal{F}} \delta\left(\mathcal{M}(x, y, \mathcal{F}_i^{(\ell)}) \neq \emptyset\right) \right] \frac{|\mathcal{M}(x, y, \mathcal{A}^{(\ell)})|}{|\mathcal{A}^{(\ell)}|}, \tag{12}$$

where δ is the indicator function of an event.

Worst case. For any input not queried yet, Alice computes the maximum of the numbers of remaining families if that input were selected next weighted by the probability of this event:

$$s^{(\ell+1)}(x) = \max_{y \in \mathcal{A}^{(\ell)}(x)} \left[\sum_{i=1}^{n_\mathcal{F}} \delta\left(\mathcal{M}(x, y, \mathcal{F}_i^{(\ell)}) \neq \emptyset\right) \right] \frac{|\mathcal{M}(x, y, \mathcal{A}^{(\ell)})|}{|\mathcal{A}^{(\ell)}|} \tag{13}$$

The input to be submitted is randomly picked among the ones with the lowest score:

$$q_{\ell+1} \in \arg\min_k s^{(\ell+1)}(x_k). \tag{14}$$

3.3 Experimental work

3.3.1 Detection

A first experimental work measures the number of queries needed for detection with the three types of family defined in Eq. (3), (4) and (5). It considers two cases: Alice's hypothesis is correct (positive case) or incorrect (negative case). The cases are not analyzed exhaustively. For example, in the negative case for a singleton family (*i.e.* Alice is wrong to suspect that the black-box is $\mathcal{F}(m, \{\theta\})$), there are $|\mathcal{A}|(|\mathcal{A}| - 1)$ possible combinations, *i.e.* more than a million. Instead, the experiment randomly picks 1.000 positive and 1.000 negative among all these cases.
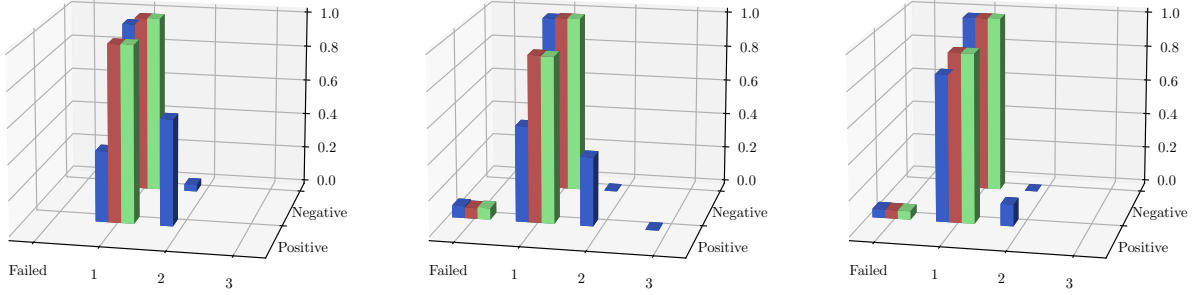
Figure 2: Probability distribution of the number of queries for $(\mathsf{D}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$ using the expectation score (7) when the black box returns top-k classes with $k = 1$ (*blue*), $k = 3$ (*red*) or $k = 5$ (*green*). Family considered from left to right: $\mathcal{F}(\mathsf{m})$ (3), $\mathcal{F}(\mathsf{m}, \Psi)$ (4), and $\mathcal{F}(\mathsf{m}, \{\theta\})$ (5).

Figure 2 shows the best results obtained with the expectation score (7). Results for the worst-case score (8) are shown in App. E.1, Fig. 14. As a side-product, Table 1 gives the percentage of inputs in $\mathcal{X}$ answering the detection problem under the best case, *i.e.* when one unique query is sufficient.

Few queries are enough.   When the detection succeeds (with a positive or negative decision about the hypothesis), at most three queries are needed, and in most cases, only one is sufficient. This holds although the greedy algorithm is known to be suboptimal. When the greedy algorithm needs three inputs, another algorithm may only need two queries. Yet, when the greedy algorithm needs two, no algorithm can do better because the greedy would have found an unique input if existing. Positive and negative conclusions are roughly drawn within the same number of queries, although our algorithm is designed to quickly prove positive detections.

Few failures.   The inability to detect the model in the black-box as part of the family $\mathcal{F}$ happens when:

$$\exists \mathsf{m} \in \mathcal{F}, \exists \mathsf{m}' \in \mathcal{A} \backslash \mathcal{F}, \forall x \in \mathcal{X}, \quad \mathsf{m}(x) = \mathsf{m}'(x). \tag{15}$$

The failures occur when the family corresponds to a set of variations (4) or an exact model (5). It happens that the algorithm cannot distinguish a few pairs of different variations issued from the same vanilla model. This is the only possible explanation: Otherwise, *i.e.* the indistinguishable models come from two different vanilla networks, this setup would also end up in a failure when detecting families spanned from a vanilla model (5), which is not reported in Fig. 2. In other words, Alice can always guess that the black-box is a variation of a given vanilla model, and rarely she cannot guess which variation it is exactly.

On the other hand, failures should also happen in the negative case. None is reported in Fig. 2 because they are statistically rare. For a given family $\mathcal{F}$, suppose that the models m and m' in (15) are both unique. A failure happens in the positive case if Bob puts model m in the black-box. This happens with probability $1/|\mathcal{F}|$. A failure happens in the negative case if Bob puts model m' in the black-box. This happens with probability $1/|\mathcal{A} \backslash \mathcal{F}| < 1/|\mathcal{F}|$.

Experimentally, the number of queries to end up in a failure is similar to the number of queries for getting a positive outcome.

A bigger top-k is better.   When the output of the black-box is rich, *i.e.* top-k classes with $k > 1$, one unique input is sufficient. Moreover, Table 1 shows that there are more of these unique inputs in $\mathcal{X}$. In this case, Alice no longer needs a large collection of benign inputs.

A bigger family is harder to detect.   Families of type (3) are bigger than families (4) which are bigger than the singleton (5). Ignoring the failure case, Figure 2 and Table 1 show that it is harder to detect a large family. It is more frequent that some model members take different outputs in large families. On the contrary,

Table 1: Percentage of inputs in $\mathcal{X}$ concluding detection $(\mathsf{D}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$ within a single query.

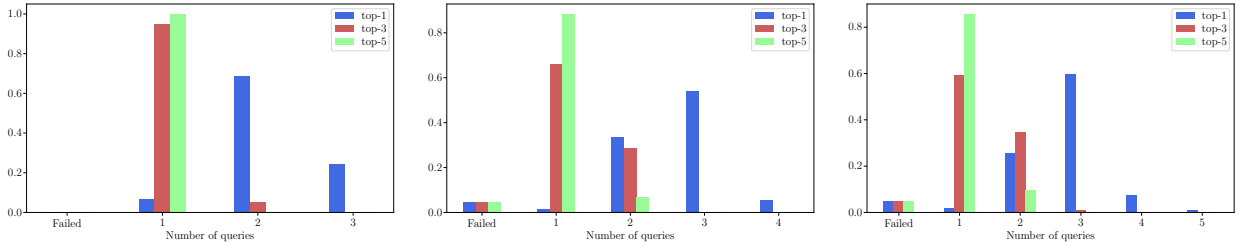| Family	top-1	top-3	top-5
Vanilla $\mathcal{F}(\mathsf{m})$ (3)	0.28%	1.2%	**12.5**%
Variation $\mathcal{F}(\mathsf{m}, \Psi)$ (4)	0.31%	4.8%	**21.0**%
Singleton $\mathcal{F}(\mathsf{m}, \{\theta\})$ (5)	0.37%	8.3%	**31.4**%

Figure 3: Probability distribution of the number of queries for $(\mathsf{I}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$ using the expectation score (7) when the black box returns top-k classes with $k = 1$ (*blue*), $k = 3$ (*red*) or $k = 5$ (*green*). Family considered from left to right: $\mathcal{F}(\mathsf{m})$ (3), $\mathcal{F}(\mathsf{m}, \Psi)$ (4), and $\mathcal{F}(\mathsf{m}, \theta)$ (5).
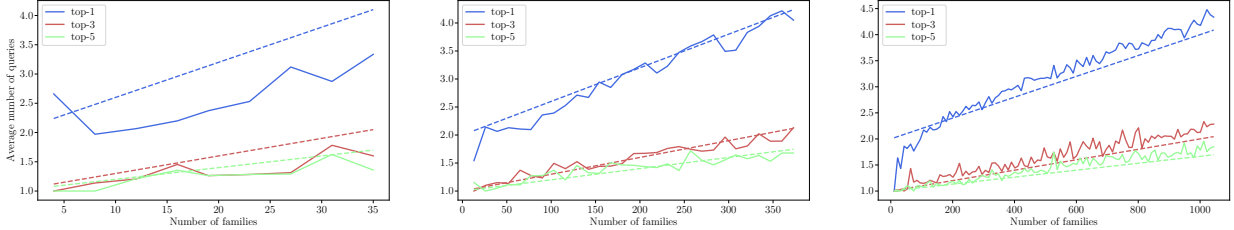

Figure 4: Average number of queries as a function of the number of families $n_{\mathcal{F}}$ for $(\mathsf{I}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$ with the expectation score (7) and when the black-box returns top-k classes with $k = 1$ (*blue*), $k = 3$ (*red*) or $k = 5$ (*green*). The dotted lines are the linear regressions. Family considered from left to right: $\mathcal{F}(\mathsf{m})$ (3), $\mathcal{F}(\mathsf{m}, \Psi)$ (4), and $\mathcal{F}(\mathsf{m}, \{\theta\})$ (5).

we observe that the variants of the same model with the same variation but with different parameters often share the same output.

3.3.2 Identification

The protocol is similar to the previous one for detection. Figure 3 shows the best results obtained with the expectation score (12). The results for the other score (13) are slightly worse and shown in App. E.1, Fig. 15.

Identification vs. Detection. Comparing Figures 2 and 3, two times more queries are necessary for identifying a family rather than detecting it. It is possible to identify a model quickly with at most five benign queries which are a lot less than the sequential procedure (10). Identification is a harder task than detection to a small extent.

The biggest difference is under the top-1 scenario where a unique query is rarely sufficient. The 35 vanilla models considered here were trained on the same dataset. They have good accuracy ($> 70\%$). If many unique inputs to identify existed, this would mean that for any of these inputs, the 35 models give 35 different top-1 predictions. Assuming that one of these models makes a correct classification, the other 34 models are wrong. If a lot of these inputs existed, this would imply models with low accuracy. In other words, these inputs are necessarily rare, or even non-existing.

A bigger top-k is better. In contrast to detection, the gain of information provided by top-3 and top-5 outputs is very substantial. When the top-5 is returned, 90% of the families are identified within one query. The supervised training of the vanilla models only focuses on the top-1 s.t. it agrees with the ground truth class. For $k > 1$, the top-k is almost specific of the network. This explains the big improvement from top-1 to top-k.

Number of families. Figure 4 represents the evolution of the average number of queries to identify one out of $n_{\mathcal{F}}$ families. The more families, the bigger the number of queries on average. But this number also depends on the size of the families and the top-k. We observe that the increase is roughly linear (see dashed lines in Fig. 4). As a rule of thumb, we observe that the expectation of the number of queries roughly follows the empirical law:

$$\mathbb{E}(L) \approx 0.002 \times \frac{\mathbb{E}(|\mathcal{F}|) n_{\mathcal{F}}}{k} + \beta(k), \tag{16}$$

8

where $\mathbb{E}(|\mathcal{F}|)$ is the average number of elements in the family. This is a major improvement w.r.t. (10). For instance, for singleton family, $\mathbb{E}(|\mathcal{F}|) = 1$ and the rate equals 0.002 under top-1, whereas the rate in (10) cannot be lower than 0.5 since we need at least one query to discard a hypothesis, *i.e.* $\mathbb{E}(L_j^{\mathsf{neg}}) \geq 1$.

4 The black-box is an unknown model

This section assumes that $\mathcal{A} \subsetneq \mathcal{B}$ because $\mathcal{B}$ contains unknown models or variants of some models known by Alice.

4.1 Modeling

4.1.1 Assumptions

Our working assumptions are the following: When queried by random inputs, a variant $\mathsf{V}(\mathsf{m}, \theta)$ produces outputs statistically

- independent from the outputs of a different model $\mathsf{m}' \neq \mathsf{m}$.

- dependent from the outputs of the original model m.

We consider a particular procedure for generating a variant as being like a transmission channel. The output Z of the variant $\mathsf{V}(\mathsf{m}, \theta)$ is as if the output Y of the original model m were transmitted to Alice through a noisy communication channel parametrized by θ. Like in C.E. Shannon's information theory of communication, we model this channel by the conditioned probabilities $W_\theta(z, y) = \mathbb{P}(Z = z | Y = y), \forall (z, y) \in \mathcal{Z}_k$.

4.1.2 Surjection

One difficulty of this context is the big size of the set $\mathcal{Z}_k$ of outcomes under the top-k assumption: $|\mathcal{Z}_k| = (C)_k$. It is then difficult to establish reliable statistics about the transition matrix W_θ which is as large as $(C)_k \times (C)_k$.

When working with top-k outputs, Alice resorts to a surjection $\mathsf{S}_k : \mathcal{Z}_k \mapsto \mathcal{S}_k$ with $\mathcal{S}_k := \{0, 1, \dots, k\}$. This greatly reduces the set of outcomes. We denote $\tilde{z} = \mathsf{S}_k(z)$ and $\tilde{y} = \mathsf{S}_k(y)$. Function S_k is slightly more complex than suggested by this simple notation. Indeed, for any input x, we assume that Alice has a reference class $c(x) \in \mathcal{C}$. If this input x is annotated, $c(x)$ is its ground truth class. Otherwise, Alice computes the top-1 output of all the models she knows, and takes a majority vote to decide on $c(x)$. For this piece of data, a model gives $\mathsf{m}(x) = (c_1, \dots, c_k)$ and the surjection makes:

$$\mathsf{S}_k(\mathsf{m}(x)) = \begin{cases} j & \text{if } \exists j : c_j = c(x) \\ 0 & otherwise. \end{cases} \tag{17}$$

In words, $\mathsf{S}_k(\mathsf{m}(x))$ is the rank of the reference class in the top-k output or 0 if the reference class is not returned. In the end, Alice uses a transmission matrix $(W_\theta(\tilde{z}, \tilde{y}))$ which is only $(k+1) \times (k+1)$.

4.2 Detection $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$

For the detection task, Alice first makes the following hypothesis: The black-box is a variant of the vanilla model $\mathsf{m} \in \mathcal{A}$. This variant may be the identity ($\mathsf{b} = \mathsf{m}$), or a variant she knows, or a variant she does not know.

Contrary to the previous section, Alice randomly chooses L inputs $(X_1, \dots, X_L) \subset \mathcal{X}$ to query the black-box and compares the observations $(\tilde{Z}_1, \dots, \tilde{Z}_L)$ to the outputs she knows $(\tilde{Y}_1, \dots, \tilde{Y}_L)$, with $\tilde{Z}_\ell := \mathsf{S}_k(\mathsf{b}(X_\ell))$, $\tilde{Y}_\ell := \mathsf{S}_k(\mathsf{m}(X_\ell)), \forall \ell \in [\![L]\!]$. We use capital letters here to outline that these are random variables since Alice randomly chooses the inputs.

There are two difficulties: i) to gauge the distance between the outputs observed from the black-box and from model m (see Sect. 4.2.1) and ii) to randomly sample informative inputs from the set $\mathcal{X}$ (see Sect. 4.2.2).

4.2.1 A discriminative distance

Alice is indeed testing two hypothesis:

- $\mathcal{H}_1$: The black-box is a variant of model m. There is a dependence between $\tilde{Z}$ and $\tilde{Y}$ which is captured by the statistical model of the variant:

$$\mathbb{P}_1(\tilde{Z} = \tilde{z}, \tilde{Y} = \tilde{y}) := W_\theta(\tilde{z}, \tilde{y}) \mathbb{P}(\tilde{Y} = \tilde{y}).$$

- $\mathcal{H}_0$: The black-box is not a variant of model m. There is no statistical dependence and
$$\mathbb{P}_0(\tilde{Z} = \tilde{z}, \tilde{Y} = \tilde{y}) := \mathbb{P}(\tilde{Z} = \tilde{z})\mathbb{P}(\tilde{Y} = \tilde{y}).$$

The well-celebrated Neyman-Pearson test is the optimal score for deciding which hypothesis holds. For L independent observations, it writes as

$$s = \sum_{j=1}^{L} \log \frac{\mathbb{P}_1(\tilde{Z} = \tilde{z}_j, \tilde{Y} = \tilde{y}_j)}{\mathbb{P}_0(\tilde{Z} = \tilde{z}_j, \tilde{Y} = \tilde{y}_j)} = \sum_{j=1}^{L} \log \frac{W_\theta(\tilde{z}_j, \tilde{y}_j)}{\mathbb{P}(\tilde{Z} = \tilde{z}_j)}. \tag{18}$$

We introduce the empirical joint probability distribution defined by

$$\hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}) := L^{-1} |\{j \in [\![L]\!] : \tilde{z}_j = \tilde{z} \text{ and } \tilde{y}_j = \tilde{y}\}| \tag{19}$$

in order to rewrite (18) as

$$s = L \sum_{(\tilde{z},\tilde{y}) \in \mathcal{S}_k^2} \hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}) \log \frac{W_\theta(\tilde{z}, \tilde{y})}{\mathbb{P}(\tilde{Z} = \tilde{z})}. \tag{20}$$

This formalization is not tractable because W_θ is not known: Alice does not know which variant θ is in the black-box, and indeed it might be an unknown variant. Yet, it guides us to a more practical score function, the empirical mutual information:

$$\hat{I}(\tilde{Z}, \tilde{Y}) := \sum_{(\tilde{z},\tilde{y}) \in \mathcal{S}_k^2} \hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}) \log \frac{\hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y})}{\hat{P}_{\tilde{Z}}(\tilde{z})\hat{P}_{\tilde{Y}}(\tilde{y})}, \tag{21}$$

with the empirical marginal probabilities:

$$\hat{P}_{\tilde{Z}}(\tilde{z}) := \sum_{\tilde{y} \in \mathcal{S}_k} \hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}), \quad \hat{P}_{\tilde{Y}}(\tilde{y}) := \sum_{\tilde{z} \in \mathcal{S}_k} \hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}). \tag{22}$$

In words, the model of the distributions $(\mathbb{P}_0, \mathbb{P}_1)$ is replaced with empirical frequencies learned on the fly. Resorting to the empirical mutual information to decode transmitted messages in digital communication is known as Maximum Mutual Information (MMI), recently proven universally optimal [10].

The empirical mutual information is a kind of similarity (the bigger, the more $\tilde{Z}$ looks like $\tilde{Y}$). Its value lies in the interval $[0, \min(\hat{H}(\tilde{Z}), \hat{H}(\tilde{Y}))]$ with the empirical entropy given by:

$$\hat{H}(\tilde{Z}) := -\sum_{\tilde{z}} P_{\tilde{Z}}(\tilde{z}) \log P_{\tilde{Z}}(\tilde{z}). \tag{23}$$

We prefer dealing with a normalized distance and we introduce:

$$D_L(\mathsf{b}, \mathsf{m}) := 1 - \frac{\hat{I}(\tilde{Z}, \tilde{Y})}{\min(\hat{H}(\tilde{Y}), \hat{H}(\tilde{Z}))} \in [0, 1]. \tag{24}$$

This defines the distance between the models b and m respectively producing $\tilde{Z}$ and $\tilde{Y}$. Indeed, Fig. 1 in the introduction is a t-SNE graphical representation extracted from such pairwise distances between models in $\mathcal{B}$. For instance, let us consider two extreme scenarios:

- The model m is in the black-box so that $\tilde{z}_j = \tilde{y}_j$, $\forall j \in [\![L]\!]$. Then $P_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}) = 1$ if $\tilde{z} = \tilde{y}$, and 0 otherwise, producing $D_L(\mathsf{b}, \mathsf{m}) = 0$.

- The black-box and model m yield independent outputs so that $P_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}) = P_{\tilde{Z}}(\tilde{z})P_{\tilde{Y}}(\tilde{y})$, then $D_L(\mathsf{b}, \mathsf{m}) = 1$.

In the end, Alice deemed the hypothesis $\mathcal{H}_1$ as being true when the distance is small enough: $D_L(\mathsf{b}, \mathsf{m}) < \tau \to \mathcal{H}_1$ is true. Alice makes two kinds of errors:

- False positive: $D_L(\mathsf{b}, \mathsf{m}) < \tau$ whereas $\mathcal{H}_1$ is false.

- False negative: $D_L(\mathsf{b}, \mathsf{m}) \geq \tau$ whereas $\mathcal{H}_1$ is true.

Alice sets the threshold τ such that the probability of false positive is lower than a required level α. The converse, *i.e.* controlling the probability of false negative, is an illusion. Appendix D shows for instance that there is no way to theoretically upper bound the distance between a variant and its original model, even if both of them share good accuracy. Our working assumption is that this mutual information is indeed large enough for a reliable hypothesis test and the experimental work confirms this in Sect. 4.4.

4.2.2 Selection of inputs

The empirical mutual information is a consistent estimator of the mutual information which depends on the channel transition matrix W_θ and the input probability distribution $P_{\tilde{Y}}$. A result of the theory of communication is that for a given transmission channel, there is an input probability which maximises the mutual information. This is of utmost importance in order to design a communication system achieving the channel capacity as defined by C.E. Shannon. In our framework, this would make the distance between a model and its variant closer to 0 likely avoiding a false negative.

However, this idea is not applicable to our scheme because Alice may know a plurality of variants, each of them leading to a different optimal input distribution. The black-box may also contain an unknown variant excluding any optimization.

Yet, when Alice randomly chooses the inputs, she has the feeling that these inputs must not be too easy to be classified otherwise any model outputs the same prediction. This is not discriminative of the presence of a given model in the black-box and it may lead to a false positive. On the other hand, these inputs must not be too hard to be classified neither otherwise the prediction tends to be random, destroying the correlation between a model and its variant. This may lead to a false negative.

The experimental work investigates several selection mechanisms of the inputs. All of them amount to randomly pick inputs from a subset $\mathcal{X}'$ of $\mathcal{X}$.

- **All.** There is indeed no selection and $\mathcal{X}' = \mathcal{X}$.

- **50/50.** Alice's hypothesis concerns a family of variants derived from a vanilla model m. $\mathcal{X}'$ is composed of 50% of inputs well classified by m (*i.e.* $\mathsf{m}(x) = c(x)$), 50% inputs for which $\mathsf{m}(x) \neq c(x)$.

- **30/70.** The same definition but with 30% well classified and 70% wrongly classified by m.

- **Entropy.** $\mathcal{X}'$ is composed of the inputs whose top-1 predictions are highly random. For a given input, Alice computes the predictions from all the models in $\mathcal{A}$ and measures the empirical entropy of these predicted labels. She then sorts the inputs of $\mathcal{X}$ by their entropy, and $\mathcal{X}'$ contains the head of this ranking.

The second and third options are dedicated to the detection task since they only need the vanilla model m at the root of Alice's hypothesis. The last selection mechanism demands a long preprocessing step depending on how big the set of models $\mathcal{A}$ is. It is dedicated to the identification task.

4.3 Identification $(\mathsf{I}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$

The identification task is nothing more than an extension of the detection. Instead of a binary hypothesis, Alice is now facing a multiple hypotheses test with $M + 1$ choices:

- $\mathcal{H}_i$: The black-box is a variant of vanilla model m_i, with $1 \leq i \leq M$,

- $\mathcal{H}_0$: The black-box is a variant of an unknown model.

The usual way is to compute distance $D_L(\mathsf{b}, \mathsf{m}_i)$ per vanilla model $\mathsf{m}_i \in \mathcal{A}$, and to decide for model $i^\star = \arg\max_{1 \leq i \leq M} D_L(\mathsf{b}, \mathsf{m}_i)$, if $D_L(\mathsf{b}, \mathsf{m}_{i^\star})$ is lower than a threshold, otherwise Alice chooses hypothesis $\mathcal{H}_0$. If a known model is in the black-box, only three events may occur:

- Alice makes a correct identification,

- Alice can not make any decision, *i.e.* she deems $\mathcal{H}_0$ as true.

- Alice makes a wrong identification.

Again, by fine-tuning the threshold, Alice controls the probability of the last event. Note that the probability of success is expected to be smaller than for the previous task. Identification is more difficult since several hypotheses are competing.

4.3.1 Compound model

Information theory helps Alice again thanks to an analogy with the communication over a compound channel. In this communication problem, a message m_i has been emitted and transmitted through a channel W_θ. The receiver knows a compound channel, *i.e.* a set of channels $\{\theta_j\}_{j=1}^{V} \subset \Theta$. It knows that the received signal has gone through one of them, but it does not know which one. There exists an optimal decoder for each channel in the set. The receiver just does not know which one to use. A theoretically grounded decoder is to decode the received signal with each of the decoders and to aggregate this decoding with a min operator [11].

The analogy is the following: the inputs go through all the models $\{m_i\}$ known by Alice, and the outputs are like messages. Bob has chosen one model, *i.e.* one of these messages. Yet, Bob uses a variant which emits noisy outputs observable to Alice. Now, suppose that Alice knows a set of variants in a given family: $\{V(m_i, \theta_j)\}_j \subset \mathcal{F}$. She uses these variants for computing distances $D_L(b, V(m_i, \theta_j))$ that she aggregates into one distance w.r.t. the family:

$$D_L(b, \mathcal{F}) := \min_j D_L(b, V(m_i, \theta_j)). \tag{25}$$

Intuitively, the black-box might be a very degraded version of a model which is indeed 'closer' to a milder variant than to the original model m_i.
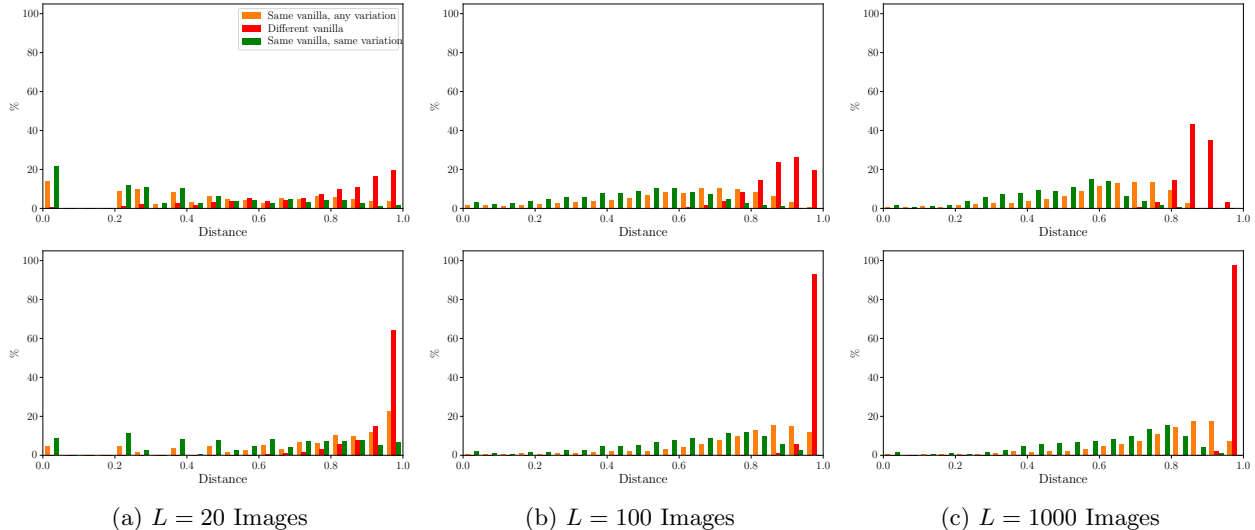

(a) $L = 20$ Images   (b) $L = 100$ Images   (c) $L = 1000$ Images

Figure 5: Histogram of the distance $D_L(m_1, m_2)$ when $(m_1, m_2) \in \mathcal{F}^2(m)$ (orange), $(m_1, m_2) \in \mathcal{F}^2(m, \Psi)$ (green), or m_1 and m_2 are variants of different vanilla models (red). Inputs randomly sampled in $\mathcal{X}$ (*top*) or in $\mathcal{X}'$ - Entropy Sect. 4.2.2- (*bottom*).

4.4 Experimental work

The previous experimental work in Sect. 3.3 considers three kinds of family concerning the black-box as defined in Sect. 2.2.2. When the family is a singleton, because $\mathcal{F} = \mathcal{F}(m, \{\theta\})$ or $\mathcal{F} = \mathcal{F}(m, \Psi)$ and $|\Psi| = 1$, then the distance between the black-box and this unique model is exactly zero. This easy case is now excluded to focus on cases where Alice does not know the variant in the black-box.

Contrary to Sect. 3.3, Alice now resorts to statistical tests. Any distance between models is a random value since the queries are randomly selected. Our protocol makes 20 measurements of any considered distance thanks to 20 independent inputs samples.

4.4.1 Assumptions about the statistical model.

Section 4.1.1 makes two assumptions about the statistical dependence between the predictions of models in the same family $\mathcal{F}$ and independence when coming from different families. Figure 5 experimentally verifies these working assumptions.

The distances between two models $V(m, \theta)$ and $V(m', \theta')$ for two vanilla models m and m' and any variants $(\theta, \theta') \in \Theta \times \Theta$ are computed. This sums up to 583,740 combinations. Figure 5 shows the histogram of these distance values over 20 bins in red. Perfect statistical independence implies a distance equaling 1. A low number of queried inputs makes the measured distance spread over a large range. The selection of the inputs has a major impact. When sampled on $\mathcal{X}$ (first row), the query may be an 'easy' input correctly classified by any model. This spoils statistical independence. When sampled on $\mathcal{X}'$ containing inputs hardly correctly classified (second row), the distances are closer to one.

The figure also shows the histogram of distances between models belonging to the same family spanned by a vanilla model m, be it $\mathcal{F}(m, \Psi)$ (same type of variation) or $\mathcal{F}(m)$ (any kind of variation). As explained in App. D, it is not possible to get a non-trivial upper bound of the distance in this case. We observe that two models from the same type of variation are usually closer. It is therefore easier to detect or identify families $\mathcal{F}(m, \Psi)$ than $\mathcal{F}(m)$.

4.4.2 Detection $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$

The experiment considers all combinations of hypothesis and model put in the black-box. There are 35 vanilla models and 1046 variants. This makes 35 families of type $\mathcal{F}(\mathsf{m})$ with an average of 30 members per family. This represents 1081 positive cases and 36,754 negative cases. There are 377 families of type $\mathcal{F}(\mathsf{m}, \Psi)$ of which 203 with a size bigger than 1. This makes 907 positive cases and 218,536 negative cases. The detection performances are gauged by the True Positive Rate (TPR) when threshold τ is set to get a False Positive Rate (FPR) of 5%.

Selection of inputs. Table 2 shows the TPR obtained when the black-box returns only *top-1* decisions. As expected, the performances for the families $\mathcal{F}(\mathsf{m}, \Psi)$ are higher. The selection Entropy is clearly the best option. Its drawback is that it needs statistics about the predictions of many vanilla models. As far as the detection task is concerned, the other selections are to be preferred because they do not require anything else than the predictions of the suspected vanilla model. In the sequel, the selection 30/70 is used for further experiments on the detection task.

The delegate model Alice measures a single distance in between the black-box and a delegate model of the hypothesis' family $\mathcal{F}$. Which member of the family is the best delegate? Three choices are proposed based on the distance to the vanilla model spanning the family: *Close*, *Median*, and *Far*. For instance, the *Close* option means that the delegate is the closest member in the family to the vanilla model:

$$\mathsf{m}_d = \arg \min_{\mathsf{m}' \in \mathcal{F}} D_L(\mathsf{m}, \mathsf{m}'). \tag{26}$$

In the case where $\mathcal{F} = \mathcal{F}(\mathsf{m})$, the closest member is m. It is not the case when $\mathcal{F} = \mathcal{F}(\mathsf{m}, \Psi)$, because the vanilla model m is not in this family. Recall that the intersection between two families has to be the empty set, otherwise Alice could not distinguish them.

Table 3 evaluates the three options. Only the 180 families with more than 3 members are considered here. For smaller families, the three options would give the same delegate.

The delegate greatly influences the results. The best choice is to select the delegate as lying at the 'center' of the family. It means the *Close* option for the family $\mathcal{F}(\mathsf{m})$, which is indeed the vanilla model m, or the *Median* option for family $\mathcal{F}(\mathsf{m}, \Psi)$.

Top-k observations The detection is evaluated for top-k outputs in Fig. 6. The best results are surprisingly obtained for $k = 1$ in Tab. 4. Our explanation is the following. The bigger k the richer the model. Yet, the empirical mutual information is calculated from $(k+1)^2$ estimated probabilities. For a given number of queries, the fewer estimations the more accurate they are. The top-1 quickly gets very good results close to 100%, so that the richer models but less accurately estimated for larger k are not competitive.

To summarize, the TPR reaches 95% for 160 queries under top-1, 200 under top-3, and 250 for top-5.

4.4.3 Identification $(\mathsf{I}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$

All conclusions obtained in the previous section are kept. Alice now has for delegate the vanilla model m for $\mathcal{F}(\mathsf{m})$ and the *Median* model for $\mathcal{F}(\mathsf{m}, \Psi)$. Images are sampled with Entropy as defined in Sect. 4.2.2.

Protocol We divide the identification task into two parts. First, Alice identifies a family $\mathcal{F}(\mathsf{m})$. As explained earlier, she can make an identification $(\mathcal{H}_i)$ or abstain $(\mathcal{H}_0)$. Depending on what is in the black-box, two correct answers are possible. In the negative case where $\mathsf{b} \in \mathcal{F}(\mathsf{m}')$ but $\mathsf{m}' \notin \mathcal{A}$, the correct answer is to abstain. The threshold τ is set s.t. the probability of error in a negative case is set to 5%. In other words, Alice minimizes her decision error when she has to abstain. For this purpose, $\mathcal{A}$ is now composed of 30 models and the remaining 5 are used to generate the negative cases. Alice thus computes the distances between b and the 30 vanilla models in $\mathcal{A}$. We repeat this 20 times where the tier of 5 excluded models is randomly sampled in $\mathcal{P}$. In the positive case where $\mathsf{b} \in \mathcal{F}(\mathsf{m}_i)$ and $\mathsf{m}_i \in \mathcal{A}$, the correct answer is to decide for hypothesis $\mathcal{H}_i$.

In the second part, Alice identifies the variation, knowing she has made a correct identification of the global family $\mathcal{F}(\mathsf{m})$. In this case, Alice has to identify the correct variation among 6 families $\{\mathcal{F}(\mathsf{m}, \Psi_j)\}_{j=1:6}$:

Table 2: True Positive Rate for $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, 1)$ with 100 random queries selected in $\mathcal{X}'$ (See Sect. 4.2.2). Delegate selected as the closest of m. False Positive Rate is set to 5%.

	All	50/50	30/70	Entropy
$\mathcal{F}(\mathsf{m})$	79.4 ± 2.1	89.2 ± 1.3	91.1 ± 1.5	$\mathbf{95.2 \pm 0.5}$
$\mathcal{F}(\mathsf{m}, \Psi)$	85.4 ± 0.9	94.1 ± 0.7	96.6 ± 0.5	$\mathbf{99.8 \pm 0.1}$

Table 3: True Positive Rate for $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$ and different delegate options with $L = 100$ random queries in 30/70, FPR = 5%.

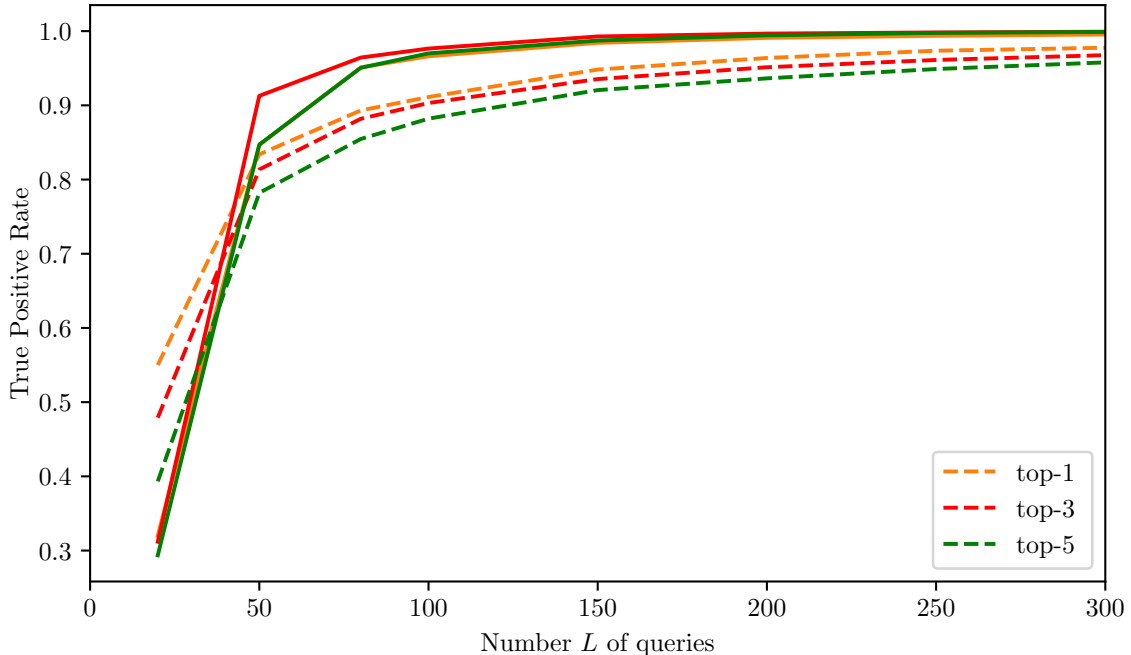| Delegate | $\mathcal{F}(\mathsf{m})$ | | | $\mathcal{F}(\mathsf{m}, \Psi)$ | | |
	top-1	top-3	top-5	top-1	top-3	top-5
Close	**91.1 ± 1.0**	**90.3 ± 1.2**	**88.2 ± 0.9**	95.5 ± 0.7	94.4 ± 0.6	92.8 ± 0.7
Median	79.5 ± 0.3	80.2 ± 0.2	78.3 ± 0.2	**96.9 ± 0.5**	**97.8 ± 0.4**	**96.9 ± 0.5**
Far	28.4 ± 2.3	33.8 ± 2.7	33.7 ± 3.5	84.8 ± 1.3	88.4 ± 1.0	97.3 ± 0.9

Figure 6: True Positive Rate for $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$ as a function of the number of queries randomly selected in 30/70, FPR = 5%, best delegate options for $\mathcal{F}(\mathsf{m})$ (dash) and $\mathcal{F}(\mathsf{m}, \Psi)$ (plain).

randomized smoothing, pruning (filter, all, last), JPEG, posterize (See App. B). Alice thus computes 6 distances based on their delegates and identifies the family $i^\star = \arg\min_j D_L(\mathsf{b}, \mathcal{F}(\mathsf{m}, \Psi_j))$. No thresholding is needed here. For each family, 20 variants with random parameters and complying with (1) are created. This leads to 700 new models tested in the black-box, different from the 1081 models considered so far.

Identifying $\mathcal{F}(\mathsf{m})$ Alice almost surely identifies the family $\mathcal{F}(\mathsf{m})$ of the black-box as shown in Fig. 7 and Tab. 5. She reaches her maximum success rate at around 300 queries. After this amount, no incorrect identification is made but 10% of abstention remains. This is due to the thresholding which prevents Alice from misclassification in the negative case. If no thresholding is done, the success rate reaches 92.8% within 100 queries and 98.2% at 500.

The number of queries is higher than for detection. For an equivalent level of performance, 4 times more queries are necessary for identification than for detection. Nevertheless, identification proceeded by sequential detection would take on average 3.000 queries (24 times more w.r.t. detection) as foreseen by (10).

Identifying $\mathcal{F}(\mathsf{m}, \Psi)$ With a single delegate, Table 5 and Figure 8 show a rather difficult identification. Variants far from the vanilla model are correctly identified. The main difficulty comes from the variations that slightly modify the model. These variants are close to m, which is the center of the cluster $\mathcal{F}(\mathsf{m})$ (see Fig. 10), therefore it is hard to distinguish them. The compound (25) with the median and the close delegates yields a boost of 12 points.

Top-k observations The best results are obtained for $k = 1$ in Tab. 5 on every task, like for detection. For the family $\mathcal{F}(\mathsf{m})$, the information gained by top-k needs too many queries to catch up with the top-1. For family $\mathcal{F}(\mathsf{m}, \Psi)$, the difference is smaller. Indeed, top-k with $k \leq 3$ gives slightly better results from ≈ 1.000 queries and above.

Table 4: True Positive Rate for $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$ with random queries selected with *30/70*, FPR = 5%.

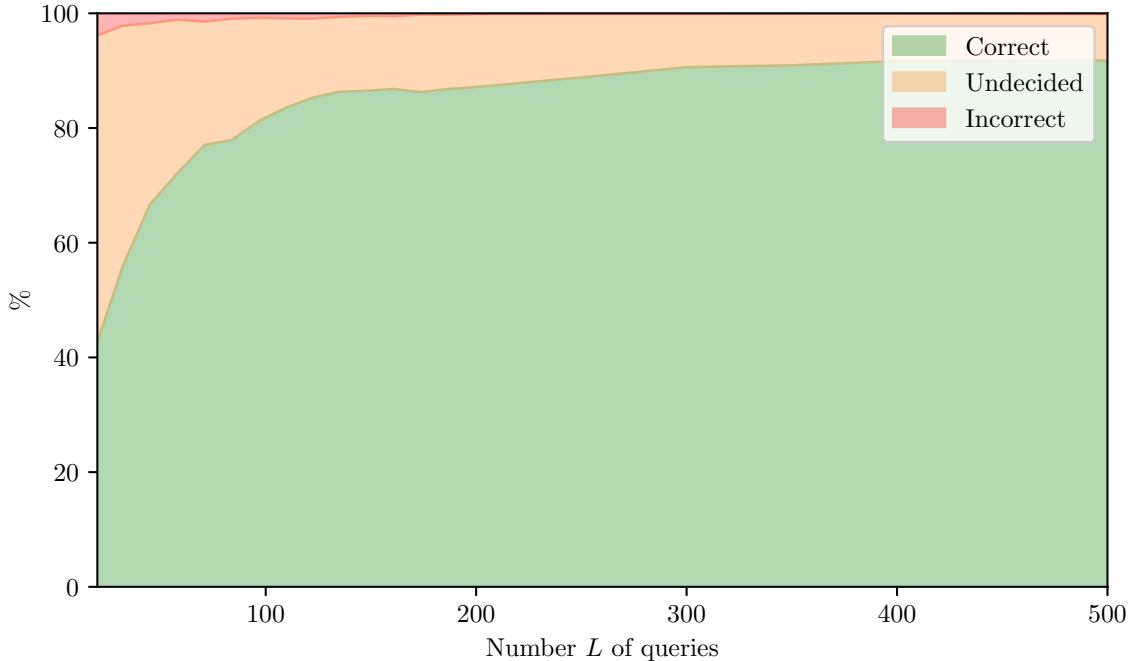|		Number of queries			
		$L = 20$	$L = 50$	$L = 100$	$L = 500$
$\mathcal{F}(\mathsf{m})$	top-1	**55.0**	**83.4**	**91.1**	**98.7**
	top-3	47.9	81.4	90.3	97.9
	top-5	39.3	78.2	88.2	97.2
$\mathcal{F}(\mathsf{m}, \Psi)$	top-1	**32.0**	84.7	96.9	99.8
	top-3	31.2	**91.3**	**97.8**	**100**
	top-5	29.4	84.7	96.9	**100**

Figure 7: Probability distribution for $(\mathsf{I}, \mathcal{F}(\mathsf{m}), \mathcal{A} \subsetneq \mathcal{B}, 1)$ vs. number L of queries. Threshold set to have a maximum 5% errors in negative cases.

5 State-of-the-art benchmark

5.1 Previous Works

Since the work of IP-Guard [2] , all the fingerprinting papers deal with adversarial examples. All the papers start with a small collection of benign inputs (except [12] starting from random noise images) and apply a precise white-box attack like CW. It forges adversarial examples s.t. they lie close to the decision boundaries, which are the signatures of a model.

Two trends are connected to two applications. The first one deals with the integrity of the model. In this scenario, Alice makes sure that Bob placed her model in the black-box without any alteration. The goal is to sense a *fragile fingerprint* s.t. any modification of the vanilla model is detectable because it changes the fingerprint. Paper [8] creates sensitive examples which are adversarial only for the vanilla model.

The second application is *robust fingerprint* as presented so far in this paper. The followers of IP-Guard [2] forge adversarial examples which are more robust in the sense that they remain adversarial for any variation of the model while being more specific to the vanilla model. Paper [3] proposes to use the universal adversarial perturbations of the vanilla model. Paper [13] introduces the concept of conferrable examples, *i.e.* adversarial examples which only transfer to the variations of the targeted model. AFA [5] activates dropout as a cheap surrogate of variants when forging adversarial examples. TAFA [4] extends this idea to other machine learning primitives like regression.

Our take in this paper is that using benign images is sufficient, and we addressed the fingerprinting problem without the need to rely on adversarial examples or any other technique to alter images to get them nearby the frontiers. Indeed, crafting adversarial examples is rather simple but forging them with extra specificities (fragile or robust to variation) is complex. It happens that all papers consider small input dimensions like MNIST or
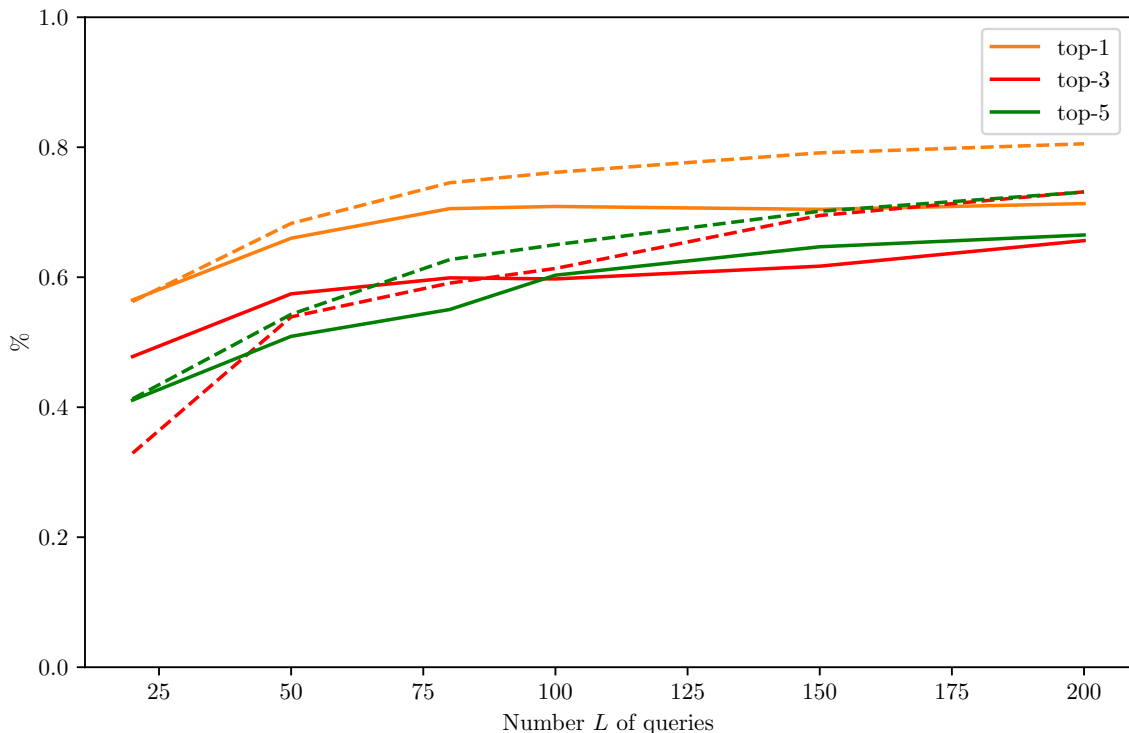
Figure 8: Correct Identification Rate for $\mathcal{F}(\mathsf{m}, \Psi)$ as a function of the number of queries. One (plain) or two (dashed) delegates per family.

CIFAR (32×32 pixel images); none of them use ImageNet (224×224) except IP-Guard [2] . Also, no paper considers that the inputs can be reformed by a defense (in order to remove an adversarial perturbation before being classified) or detected as adversarial [14].

5.2 Fragile fingerprinting

The application considered in [8] imagines that Alice wants to detect whether the black-box is exactly m and not a variant. This corresponds to our scenario $(\mathsf{D}, \mathcal{F}(\mathsf{m}, \{\theta\}), \mathcal{A} = \mathcal{B}, 1)$ where θ is the identity variation, and $\mathcal{A} = \mathcal{F}(\mathsf{m})$.

We create $L = 20$ sensitive examples per model with 200 iterations and two distortion budgets ($\epsilon = 8/255$ and $16/255$) using their code[2]. It happens that its performance on ImageNet (reported in Tab. 6) is lower than the one reported in [8] on small input size datasets (like CIFAR). Especially, this scheme can not distinguish the vanilla model and its variants 'JPEG' or 'Half precision' even with a number of queries ($L = 20$) bigger than the one recommended ($L = 8$) in [8]. Our scheme needs no more than *two* queries and performs perfectly except when pruning the last layer for five out of thirty-five models.

5.3 Robust fingerprinting

This application is related to our scenario $(\mathsf{D}, \mathcal{F}(\mathsf{m}), \mathcal{A} \subsetneq \mathcal{B}, k)$ IP-Guard [2] is the only work showing to be tractable and effective on large input size like in ImageNet. It leverages several white-box attacks to create adversarial examples. The best results demonstrated in the paper are with the attack CW [15]. We instead use BP [16] because it exhibits similar performances while being much faster (only 50 iterations). The BP implementation is from GitHub[3].

Table 7 compares the performances under 100 and 200 queries and top-1 observations. Any selection of the inputs beats IP-Guard [2] . Detailed results are reported in App. E.3. Some variations are easier to be detected ('precison', 'pruning') and the two methods are on par. On the contrary, randomized smoothing, a variation never considered in the literature, is more difficult. IP-Guard [2] is based on crafting adversarial examples close to the decision boundaries which are greatly 'crumpled' by random smoothing. Not relying on adversarial examples seems to be a clear advantage here. Our method offers more stability in the results: No variation pulls the TPR below 85%.

---

[2]Sensitive Examples' GitHub: `https://github.com/zechenghe/Sensitive_Sample_Fingerprinting`
[3]Boundary Projection's GitHub: `https://github.com/hanwei0912/walking-on-the-edge-fast-low-distortion-adversarial-examples`

Table 5: Correct Identification Rate for $(\mathsf{I}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$ with random queries selected with Entropy.

		Number of queries		
		$L = 50$	$L = 100$	$L = 500$
$\mathcal{F}(\mathsf{m})$ delegate = {close}	top-*1*	**66.7**	**81.3**	**91.8**
	top-*3*	22.6	37.6	79.1
	top-*5*	21.1	48.0	82.0
$\mathcal{F}(\mathsf{m}, \Psi)$ delegate = {median}	top-*1*	66.0	70.9	73.7
	top-*3*	57.4	59.7	71.5
	top-*5*	50.9	60.3	69.8
$\mathcal{F}(\mathsf{m}, \Psi)$ delegate = {close, median}	top-*1*	**68.3**	**76.1**	**82.5**
	top-*3*	53.9	61.4	80.6
	top-*5*	54.3	65.0	80.5

Table 6: False Positive Rate for the $(\mathsf{D}, \mathcal{F} = \{\mathsf{m}\}, \mathcal{A} = \mathcal{B}, 1)$ task.

		Finetuning		Histo.	Random.	Prune			Poster.	Half	JPEG
		All	Last		Smooth.	All	Last	Filter		Prec.	
Sens. Ex. [8] $L = 20$	$\epsilon = \frac{8}{255}$	18.1	20.3	0	35.7	15.4	31.5	57.4	77.0	100	100
	$\epsilon = \frac{16}{255}$	21.4	1.8	0	31.0	10.0	21.4	48.5	63.4	97.1	97.1
FBI with $L = 2$		**0**	**0**	**0**	**0**	**0**	**16.9**	**0**	**0**	**0**	**0**

6 Related work

Model watermarking Having reviewed the related work dedicated to the fingerprinting task in the previous section, we here review another closely related domain: the *watermarking* of models.

Watermarking is the active counterpart of fingerprinting: instead of relying on specifics of a fixed model to devise its unique fingerprint, watermarking modifies the model for which ownership must be proven. While watermarking is a common practice for decades in the field of image processing [17], it has just recently been incepted into the machine learning domain. Uchida et al. [18] first proposed to watermark a deep neural network by embedding it into the weights and biases of the model. Quickly after this initial proposal, works instead focused on a black-box model, where the presence of a watermark can be assessed by Alice from remote interaction with the suspected deep neural network, just like for fingerprinting. In [19], authors insert information by altering the decision boundaries through finetuning. In [20], authors also retrain the model to obtain wrong labels for a so-called *trigger set* of inputs, that constitutes the watermark. Please refer to [21] for a complete overview of the domain.

Some papers claim that robust fingerprinting could replace watermarking with the clear advantage that no modification of the model is needed [5]. We strongly disagree. No fingerprinting scheme, included ours, brings any formal guarantee on what is in the black-box, just like for human biometric traits. It is indeed the primary goal of watermarking to guarantee a certain level of trust.

7 Conclusion

The problem of accurate and efficient fingerprinting of valuable modern models is salient. This paper demonstrates that such a demand can be fulfilled by just using benign inputs, in not only the classic detection task,

Table 7: Comparison of True Positive Rates for the $(\mathsf{D}, \mathcal{F}(\mathsf{m}), \mathcal{A} \subsetneq \mathcal{B}, 1)$ task. FPR set to 5%.

Fingerprinting scheme	Parameters	Number of queries	
		$L = 100$	$L = 200$
IP-Guard [2]	BP [16] & 50 iter.	66.9	72.7
FBI	Random	79.4	91.5
	30/70	91.1	97.4
	Entropy	**95.2**	**97.6**

but also in the novel identification task we have introduced. This implies that we no longer need models in white-box access to compute their fingerprints.

We outline the following lessons:

In the walled garden setup of Sect. 3, less than ten inputs are needed but these are sequentially and carefully selected among a large collection depending on the previous outputs of the black-box. In other words, the key is the interaction between the greedy algorithm and the black-box. Observing top-1, top-3 or top-5 makes a difference. It is easier to spot inputs that single out a model with richer outputs.

In the open-world setup of Sect. 4, hundreds of inputs are necessary but the scheme is not iterative and selection is less crucial. Surprisingly, observing richer outputs does not yield a gain in this setup.

The identification task is more complex than detection but by a small amount only. Our identification schemes are much more efficient than the naive sequential search.

The hardest variation of our experimental protocol is randomized smoothing for robust fingerprinting and pruning the last layer for fragile fingerprinting. It means that the former reduces the statistical dependence of the outputs while the latter hardly perturbs the outputs given by the vanilla model.

One limitation of our work is that it cannot handle classifiers whose accuracy is almost perfect. This would happen for too easy classification setups where the value of models is lower, and fingerprinting is less critical. We nevertheless expect future models and applications to be complex tasks, where reaching good accuracy levels will remain a struggle.

References

[1] Si Wang and Chip-Hong Chang. Fingerprinting deep neural networks - a deepfool approach. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2021.

[2] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. Association for Computing Machinery, 2021.

[3] Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations, 2022.

[4] Xudong Pan, Mi Zhang, Yifan Lu, and Min Yang. Tafa: A task-agnostic fingerprinting algorithm for neural networks. In *European Symposium on Research in Computer Security*, pages 542–562. Springer, 2021.

[5] Jingjing Zhao, Qingyue Hu, Gaoyang Liu, Xiaoqiang Ma, Fei Chen, and Mohammad Mehedi Hassan. Afa: Adversarial fingerprinting authentication for deep neural networks. *Computer Communications*, 150:488–497, 2020.

[6] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[8] Zecheng He, Tianwei Zhang, and Ruby Lee. Sensitive-sample fingerprinting of deep neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017.

[10] Ran Tamir and Neri Merhav. The mmi decoder is asymptotically optimal for the typical random code and for the expurgated code, 2020.

[11] Emmanuel Abbe and Lizhong Zheng. Linear universal decoding for compound channels. *IEEE Transactions on Information Theory*, 56(12):5999–6013, 2010.

[12] Siyue Wang, Pu Zhao, Xiao Wang, Sang Chin, Thomas Wahl, Yunsi Fei, Qi A Chen, and Xue Lin. Intrinsic examples: Robust fingerprinting of deep neural networks. In *British Machine Vision Conference (BMVC)*, 2021.

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

[14] Anouar Kherchouche, Sid Ahmed Fezza, Wassim Hamidouche, and Olivier Déforges. Detection of adversarial examples in deep neural networks with natural scene statistics. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2020.

[15] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[16] Benoit Bonnet, Teddy Furon, and Patrick Bas. Generating Adversarial Images in Quantized Domains. *IEEE Transactions on Information Forensics and Security*, 2022.

[17] M.D. Swanson, Bin Zhu, and A.H. Tewfik. Transparent robust image watermarking. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 211–214 vol.3, 1996.

[18] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 269–277, 2017.

[19] Erwan Le Merrer, Patrick Perez, and Gilles Trédan. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 2020.

[20] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1615–1631, 2018.

[21] Yue Li, Hongxia Wang, and Mauro Barni. A survey of deep neural network watermarking techniques. *Neurocomputing*, 461:171–193, 2021.

[22] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[23] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*. Association for Computing Machinery, 2010.

[24] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. Barrage of random transforms for adversarially robust defense. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[25] Uri Shaham, James Garritano, Yutaro Yamada, Ethan Weinberger, Alex Cloninger, Xiuyuan Cheng, Kelly P. Stanton, and Yuval Kluger. Defending against adversarial images using basis functions transformations. *CoRR*, abs/1803.10840, 2018.

[26] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

[27] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E. Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*, KDD '18. Association for Computing Machinery, 2018.

[28] Anibal Pedraza, Oscar Deniz, and Gloria Bueno. Approaching adversarial example classification with chaos theory. *Entropy*, 22(11), 2020.

[29] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *Proceedings 2018 Network and Distributed System Security Symposium*, 2018.

[30] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

[31] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *S&P*, 2019.

[32] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Second-order adversarial attack and certifiable robustness. 2019.

[33] Thibault Maho, Teddy Furon, and Erwan Le Merrer. Randomized Smoothing under Attack: How Good is it in Pratice? In *ICASSP 2022 - IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE.

[34] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets, 2016.

[35] Hanwei Zhang, Yannis Avrithis, Teddy Furon, and Laurent Amsaleg. Walking on the Edge: Fast, Low-Distortion Adversarial Examples. *IEEE TIFS*, 2020.

A Notations

Notations are summarized in Tab. 8.

Table 8: Notations

D	the task of detection
I	the task of identification
m	a vanilla model as listed in Tab. 9
$V(m, \theta)$	variant obtained by applying procedure θ on m
Θ	set of all variation procedures and parameters
$acc(m)$	accuracy of model m
b	the model in the black-box
z	top-k output of the black-box
$\mathcal{C}$	the set of classes labeled from 1 to C
$\mathcal{Z}_k$	the set of top-k outcomes
$(C)_k$	falling factorial $(C)_k := C(C-1)\dots(C-k+1)$
$\mathcal{B}$	set of models Bob can create as defined in (2)
$\mathcal{A}$	set of models known by Alice
$\mathcal{P}$	set of public vanilla models listed in Tab. 9
$\mathcal{F}(m)$	family spanned by vanilla model m (3)
$\mathcal{F}(m, \Psi)$	family spanned by m and variation $\Psi \subset \Theta$ (4)
$\mathcal{X}$	a collection of benign inputs
N	size of $\mathcal{X}$
$q_{1:\ell}$	an ordered list of indices in $[\![N]\!]$
$[\![N]\!]$	integers from 1 to N
$\mathcal{D}(x)$	set of outputs given by models in $\mathcal{D}$ for input x
$\mathcal{M}(x, y, \mathcal{D})$	subset of models of $\mathcal{D}$ giving y for input x
$(\mathcal{A}\backslash\mathcal{F})^{(\ell)}$	subset of candidate models at step ℓ (6)
$s^{(\ell+1)}(x)$	score of input x at step $\ell+1$ when $\mathcal{A} = \mathcal{B}$
L^{pos}, L^{neg}	nb. of queries for a positive / negative detection
S_k	surjection from $\mathcal{Z}_k$ to $\mathcal{S}_k := \{0, 1, \dots, k\}$
z, y	top-k output of black-box b or of model m
Z, Y	random top-k output when the input is random
$\tilde{z}, \tilde{Z}, \tilde{y}, \tilde{Y}$	similar outputs after the sujection
W_θ	$(k+1) \times (k+1)$ transition matrix
$\hat{I}(\tilde{Z}, \tilde{Y})$	Empirical mutual information in bits
$\hat{H}_{\tilde{Z}}(\tilde{z})$	Empirical entropy in bits
$D_L(m_1, m_2)$	Distance between models with L queries
$D_L(b, \mathcal{F})$	Distance of the black-box from family $\mathcal{F}$

B Models and variations

B.1 Description of the set of models

The core of the set $\mathcal{A}$ contains 35 'off-the-shelves' vanilla models which were trained with supervision for the task ImageNet over one million annotated 224×224 pixel images. These models come from the Timm [22] and Torchvision [23] libraries. Among them, some are very close. For instance, there are 6 versions of EfficientNet-b0. These models share the same architecture but they result from different training session. Yet, they are considered as different models and this proves the efficiency of our method.

To forge a variant, we have selected 8 transformations. These are simple procedures easily applicable by Bob. They move the decision boundary of the model with a limited drop in accuracy. As such, all of them have already proven themselves as a defense against adversarial examples [24].

- **Identity**: The variant is an exact copy of the model.

- **Model Precision**: Deep neural networks usually encode weights and biases on 32 bits floating point precision. The Torch class attribute `half` is used to reduce the precision to 16 bits.

- **JPEG Compression**: Before being classified, the input image goes through a JPEG compression. This has been proposed as a defense: JPEG coarsely quantizes the high frequencies while adversarial perturbations are essentially composed of high frequencies. JPEG compression and decompression act as a reformer [25, 26, 27]. No training is needed because models are robust to JPEG compression. The quality factor ranges from 30 to 90 in step of 10.

- **Histogram Equalizer**: This increases the contrast in an image and it has been proposed as a defense against adversarial perturbation in [24, 28]. We use the function `transforms.functional.equalize` from Torchvision.

- **Color Depth Reduction**: Another defense is to reduce the depth of color channels to less than 8 bits [29], here from 3 and 7 bits. The image is posterized with the function `transforms.functional.posterize` from Torchvision.

- **Randomized Smoothing**: Randomized smoothing provides robustness guarantees [30, 31, 32] and is also efficient against black-box attacks [33]. We take the Github implementation[4] of [30]. The number of samples is set to 100 as recommended in the previous works. The standard deviations σ selected are: 0.01, 0.02, 0.04, 0.06, 0.08, 0.1.

- **Finetuning**: Finetuning updates the weights of the model during a new training. We consider finetuning all the layers or only the last one. Finetuning runs over 50 epochs with SGD optimizer.

- **Pruning**: Pruning compresses the model by removing the less important weights. It is applied with the function `nn.utils.prune.l1_unstructured` from Torch package. It removes the weights with the lowest ℓ_1 norm. Pruning can be applied on all the layers or just some particular ones. Filter pruning [34] cuts the less important output channels of the convolutional layers. We consider three options with the following fraction of weights removed:

    - Pruning All layers: 1%, 2%, 3%, 4%
    - Filter Pruning: 10%, 20%, 30%,
    - Pruning the Last layer: 70%, 80%, 90%, 95%

This makes a total of 33 procedures, most of them being easily and quickly applicable to any vanilla model. Few of them imply a light retraining which is done with a subset of 50,000 images of the ImageNet validation set. We end up with 1189 variants. The accuracy of each of them is measured on the remaining part of the ImageNet validation set. If the drop of accuracy is bigger than 15% compared to the original model, then the variant is discarded. Our final collection contains 1081 models and variants.

Alice has a collection of 20.000 images randomly taken from the ImageNet test set. These images are not annotated with a ground truth. These were not used for training the models or retraining the variants.

B.2 Distance between models

The distances between all the pairs of models is shown in Fig. 9. The block diagonal shows that the distances between variants of the same vanilla models are small. This distance matrix is the input of the t-SNE algorithm which creates the 2D representation of Fig. 11. We clearly see the cluster of variants centered on each vanilla model.

---

[4]Randomized Smoothing GitHub: https://github.com/locuslab/smoothing

Table 9: List of the 35 vanilla models and their variants. *all* means that all values listed in App. B.1 comply with (1).

	Nb of variants	Finetuning All	Finetuning Last	Half Precision	Histogram	JPEG	Posterize	Prune All	Prune Filter	Prune Last	Randomized Smoothing
$CoaT_{LiteSmall}$	29	✓	✓	✓	✓	all	all	0.01	all	0.7, 0.8, 0.9	all
$ConViT_{small}$	28	✓	✓		✓	all	all	0.01	all	0.7, 0.8, 0.9	all
DLA_{102}	30	✓	✓	✓		all	4, 5, 6, 7	all	all	all	0.01, 0.02, 0.04, 0.06, 0.08
DLA_{60}	32	✓	✓	✓		all	all	all	all	all	all
DLA_{68b}	32	✓	✓	✓	✓	all	all	0.01, 0.02, 0.03	all	all	all
DPN_{92}	33	✓	✓	✓	✓	all	all	all	all	all	all
(Torch) $EfficientNet_{b0}$	28	✓	✓	✓	✓	all	all	0.01, 0.02	0.1, 0.2	0.7, 0.8, 0.9	0.01, 0.02, 0.04, 0.06, 0.08
(TF) $EfficientNet_{b0}$	25	✓	✓	✓	✓	all	all	0.01	0.1, 0.2	0.7, 0.8	0.01, 0.02, 0.04
(TF) $EfficientNet_{b0,AP}$	28	✓	✓	✓	✓	all	all	0.01, 0.02	0.1, 0.2	0.7, 0.8	all
(TF) $EfficientNet_{b0,NS}$	28	✓	✓	✓	✓	all	all	0.01, 0.02	0.1, 0.2	0.7, 0.8, 0.9	0.01, 0.02, 0.04, 0.06, 0.08
(TF) $EfficientNetV2_{b0}$	30	✓	✓	✓	✓	all	all	0.01, 0.02, 0.03	all	0.7, 0.8, 0.9	0.01, 0.02, 0.04, 0.06, 0.08
$HRNet_{w30}$	31	✓	✓	✓		all	all	all	all	all	0.01, 0.02, 0.04, 0.06, 0.08
$LeViT_{128s}$	28		✓	✓	✓	all	all	0.01	0.1, 0.2	all	all
$LeViT_{256}$	30	✓	✓	✓	✓	all	all	0.01, 0.02	all	all	all
$LeViT_{384}$	32	✓	✓	✓	✓	all	all	0.01, 0.02, 0.03	all	all	all
(Torch) $ResNet50$	28	✓	✓	✓		all	4, 5, 6, 7	all	all	0.7, 0.8, 0.9	0.01, 0.02, 0.04, 0.06
$ResNet50_{madry}$	30	✓	✓	✓		all	all	all	all	0.7, 0.8, 0.9	0.01, 0.02, 0.04, 0.06, 0.08
$MixNet_l$	29	✓	✓	✓	✓	all	4, 5, 6, 7	0.01, 0.02, 0.03	all	0.7, 0.8, 0.9	0.01, 0.02, 0.04, 0.06, 0.08
$MixNet_s$	25	✓	✓	✓		all	4, 5, 6, 7	0.01, 0.02	0.1, 0.2	0.7, 0.8, 0.9	0.01, 0.02, 0.04, 0.06
$MixNet_{xl}$	32	✓	✓	✓	✓	all	all	all	all	0.7, 0.8, 0.9	all
$MobileNetV2_{120d}$	27	✓	✓	✓	✓	all	all	0.01, 0.02, 0.03	0.1, 0.2	0.7, 0.8	0.01, 0.02, 0.04, 0.06
$MobileNetV2_{140}$	25	✓	✓	✓	✓	all	all	0.01, 0.02	0.1	0.7, 0.8	0.01, 0.02, 0.04, 0.06
$MobileNetV3_{rw}$	25	✓	✓	✓		all	4, 5, 6, 7	0.01, 0.02, 0.03	0.1, 0.2	0.7, 0.8	0.01, 0.02, 0.04, 0.06
$PiT_{s,distilled}$	33	✓	✓	✓	✓	all	all	all	all	all	all
$RegNetX_{32}$	31	✓	✓	✓		all	all	all	all	0.7, 0.8, 0.9	all
$ResNet50$	32	✓	✓	✓	✓	all	all	0.01, 0.02, 0.03	all	all	all
$BiT_{R50\times1}$	32	✓	✓	✓	✓	all	all	0.01, 0.02, 0.03	all	all	all
$ReXNet_{130}$	30	✓	✓	✓	✓	all	all	0.01, 0.02, 0.03	0.1, 0.2	0.7, 0.8, 0.9	all
$ReXNet_{150}$	31	✓	✓	✓	✓	all	all	0.01, 0.02, 0.03	0.1, 0.2	all	all
$ReXNet_{200}$	31	✓	✓	✓	✓	all	all	0.01, 0.02, 0.03	0.1, 0.2	all	all
$Swin_T$	28	✓	✓	✓	✓	all	all		all	0.7, 0.8, 0.9	all
$ResNeXt_{101,32x4d}$	33	✓	✓	✓	✓	all	all	all	all	all	all
$ResNeXt_{50,32x4d}$	33	✓	✓	✓	✓	all	all	all	all	all	all
$Twins-PCPVT_{base}$	28	✓	✓	✓	✓	all	all		all	0.7, 0.8, 0.9	all
$ViT_{base,patch=16}$	29	✓	✓	✓	✓	all	all	0.01, 0.02	all	0.7, 0.8	all

C Proof of Equation (10)

Suppose that Bob selects $b \in \mathcal{F}_j$. Alice makes a random permutation σ of the $n_{\mathcal{F}}$ families and sequentially tests them. Then, she spends L_i^{neg} queries to discover that the black-box does not belong to $\mathcal{F}_i$, for any i s.t. $\sigma(i) < \sigma(j)$ (*i.e.* the families ranked by the permutation before $\mathcal{F}_j$) and L_j^{pos} queries to discover that the black-box is a member of $\mathcal{F}_j$. Over all the random permutations, there is statistically one chance out of two that $\mathcal{F}_i$ is ranked before $\mathcal{F}_j$. On expectation, this makes the following number of queries:

$$\mathbb{E}\left(L_j^{\mathsf{pos}}\right) + \frac{1}{2}\sum_{i \neq j}\mathbb{E}\left(L_i^{\mathsf{neg}}\right) = \mathbb{E}\left(L_j^{\mathsf{pos}}\right) + \frac{1}{2}\left(-\mathbb{E}\left(L_j^{\mathsf{neg}}\right) + \sum_{i=1}^{n_{\mathcal{F}}}\mathbb{E}\left(L_i^{\mathsf{neg}}\right)\right).$$

We now suppose that Bob select a family uniformly at random, to obtain the following average:

$$\mathbb{E}(L) = \frac{1}{n_{\mathcal{F}}}\sum_{j=1}^{n_{\mathcal{F}}}\mathbb{E}\left(L_j^{\mathsf{pos}}\right) + \frac{n_{\mathcal{F}} - 1}{2n_{\mathcal{F}}}\sum_{j=1}^{n_{\mathcal{F}}}\mathbb{E}\left(L_j^{\mathsf{neg}}\right). \tag{27}$$

D Lower bound of the distance D_L

This section provides an example of the computation of the mutual information between the outputs of model m and the black-box b. It considers that these classifiers yield only top-1 outputs. We assume that the surjection S_1 is defined in (17) w.r.t. the ground truth: $\tilde{z} = 1$ when the black-box succeeds in predicting the ground truth. The joint probability distribution is denoted by:

$\mathbb{P}(\tilde{Z}, \tilde{Y})$	$\tilde{Y} = 0$	$\tilde{Y} = 1$
$\tilde{Z} = 0$	a	b
$\tilde{Z} = 1$	c	$1 - a - b - c$

We suppose that the accuracies of both models are known, $\mathsf{acc}(m) = A$ and $\mathsf{acc}(b) = B$, so that:

$$\mathsf{acc}(m) = \mathbb{P}(\tilde{Y} = 1) = 1 - a - c = A, \tag{28}$$
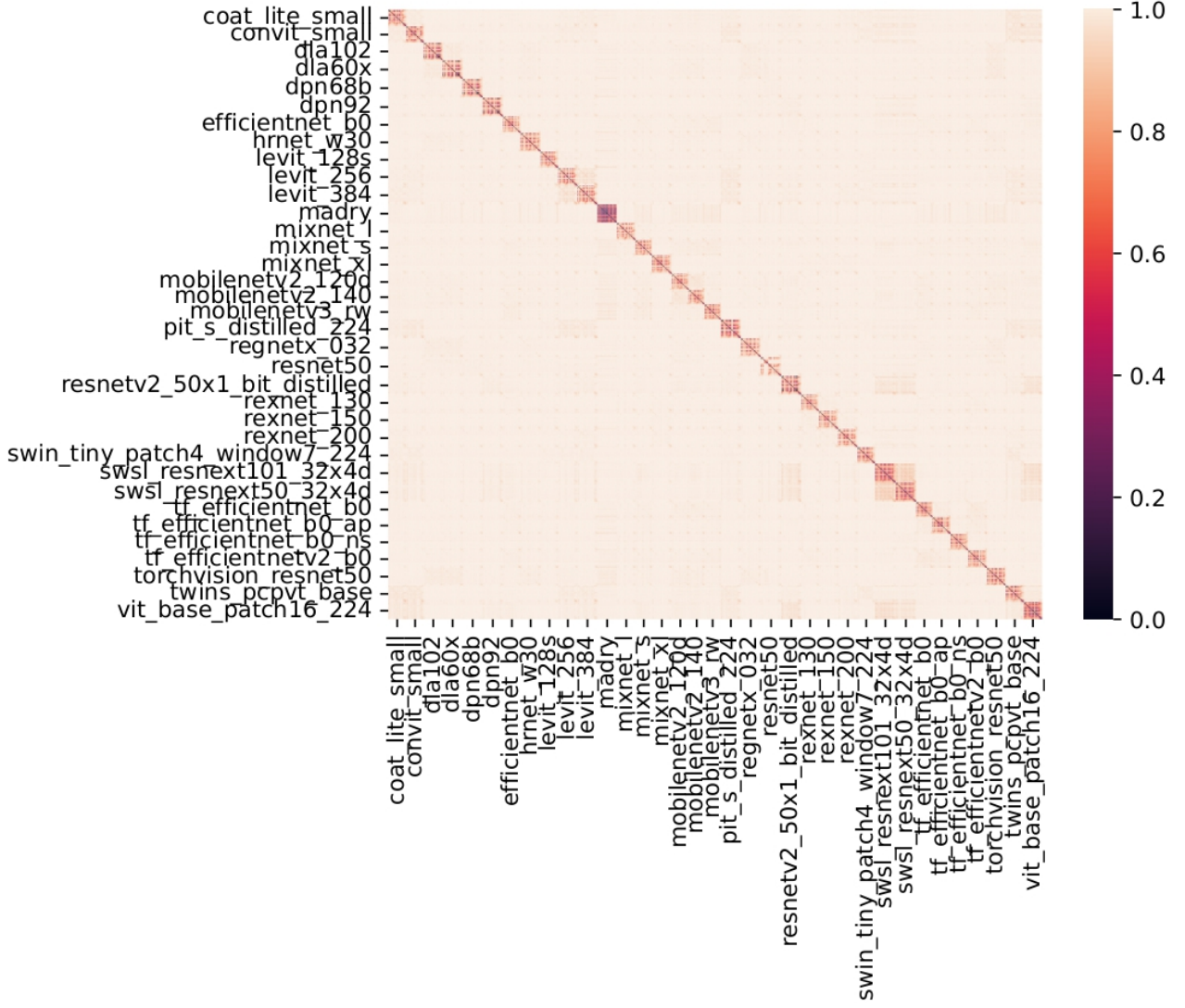$$\mathsf{acc}(b) = \mathbb{P}(\tilde{Z} = 1) = 1 - a - b = B. \tag{29}$$

Figure 9: Distances between pairs of models.

These equations are constraints reducing the problem with three unknown parameters (a, b, c) to a single one: From a, we can easily deduce (b, c) from the above equations. Since all these joint probabilities are between 0 and 1, this implies that

$$\max(0, 1 - (A + B)) \leq a \leq \min(1 - A, 1 - B). \tag{30}$$

The mutual information between the outputs $\tilde{Z}$ and $\tilde{Y}$ is given by:

$$
\begin{aligned}
I(\tilde{Z}; \tilde{Y}) &= H(\tilde{Z}) + H(\tilde{Y}) - H(\tilde{Z}, \tilde{Y}) \geq 0 \\
&= h(A) + h(B) - \sum_{(\tilde{z}, \tilde{y}) \in \{0,1\}^2} f(\mathbb{P}(\tilde{Z} = \tilde{z}, \tilde{Y} = \tilde{y}))
\end{aligned}
\tag{31}
$$

where $f(x) := -x \log_2 x$ and $h(p) := f(x) + f(1 - x)$ is the binary entropy in bits for $p \in [0, 1]$. Thanks to the constraints (28) and (29), the mutual information is a function of a, say $I(\tilde{Z}; \tilde{Y}) = I(a)$. Its derivative cancels at only one value giving $I(a) = 0$ achieved when $\tilde{Z}$ and $\tilde{Y}$ are independent, i.e.

$$a = (1 - A)(1 - B), \quad b = A(1 - B), \quad c = (1 - A)B, \quad d = AB.$$

This is thus a global minimum and the maximum of $I(\tilde{Z}; \tilde{Y})$ lies on the boundary of the range of a. Suppose that $A \geq B$ and $A + B > 1$, then

$$I(\tilde{Z}; \tilde{Y}) \leq f(A) + \max\left(f(B) - f(A + B - 1), f(1 - B) - f(A - B)\right)$$

which is converted into a lower bound of the distance $D_L(\mathsf{b}, \mathsf{m})$

$$1 - \frac{f(A) + \max\left(f(B) - f(A + B - 1), f(1 - B) - f(A - B)\right)}{\min(f(A) + f(1 - A), f(B) + f(1 - B))}.$$
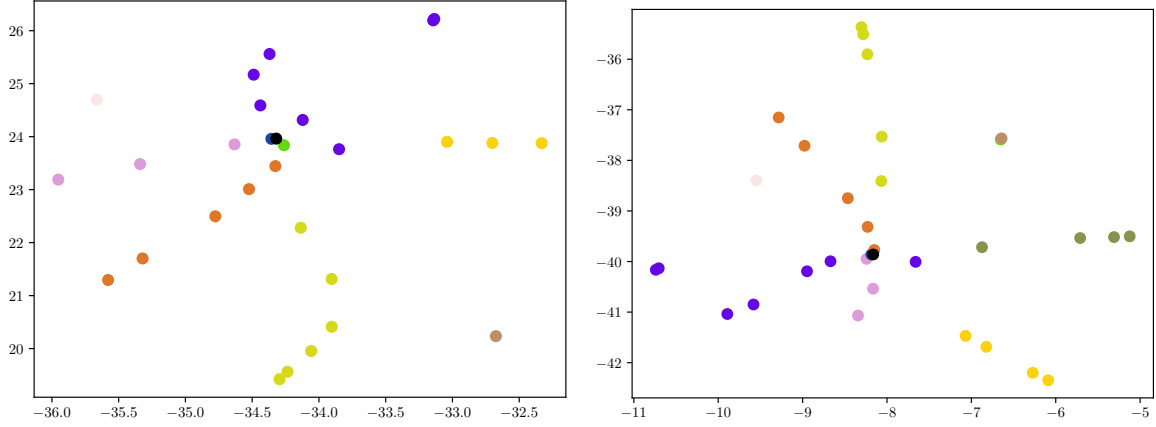
24

Figure 10: t-SNE representation of a single model and his variation. Each color is a variation (see color code of Fig. 11). The selected vanilla models are $Swin_T$ on the left and $RegNetX_{32}$.
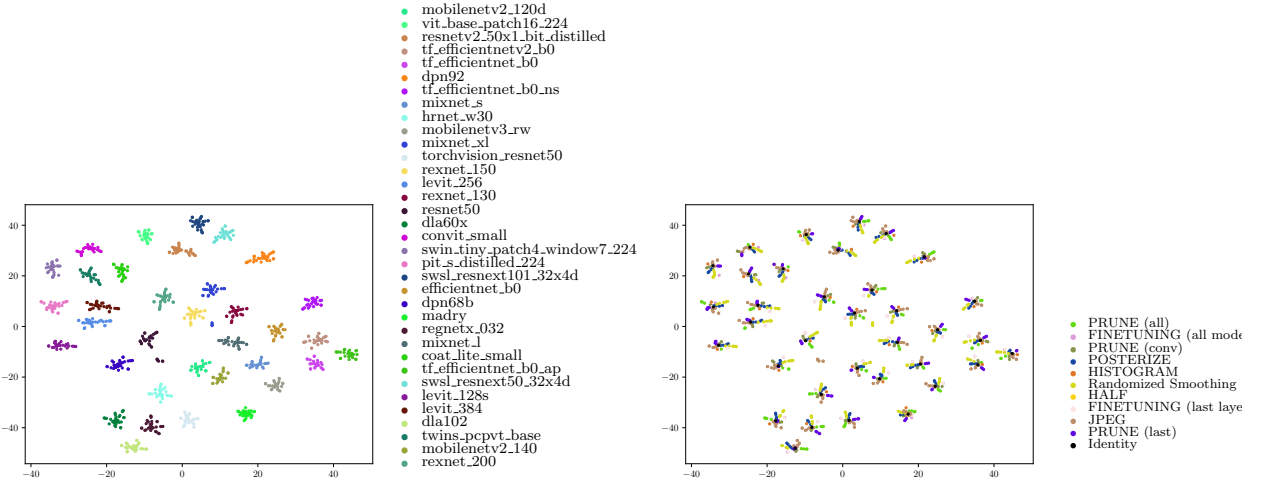

Figure 11: t-SNE representation on the distances between models. On the left, each color is associated to a vanilla model. On the right, the same representation with a different color code: the color now represents the variation applied to the vanilla models.

This function is illustrated in Fig. 12 where we see that $D_L(\mathsf{b}, \mathsf{m})$ may cancel only when $A = B$ or $A = 1 - B$. Figure 13 compares this lower bound with the actual measurements for $A = 0.45$.

E   Supplementary results

E.1   Known model $\mathcal{A} = \mathcal{B}$

Figures 14 and 15 show detection and identification results when using worst-case score (8).

E.2   Unknown model $\mathcal{A} \subset \mathcal{B}$

Table 10 justifies the use of a mixture of well and misclassified inputs under $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, 1)$.

E.3   Benchmark

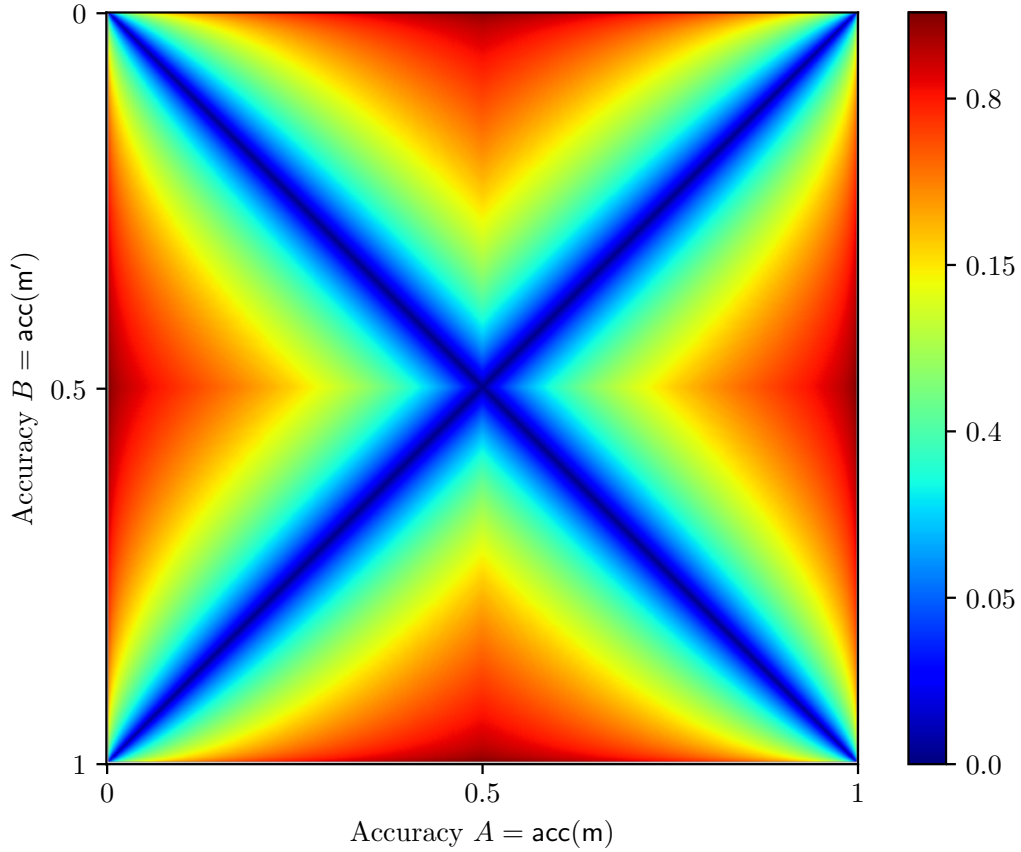Table 11 details the true positive rates per variation.

Figure 12: Lower bound of the distance between two models of accuracy A and B under top-1.

Table 10: True Positive Rate for $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, 1)$ when queried inputs are a mixture of well or misclassified images. False Positive Rate is set to 5%.

| Well / Mis-classified	50 inputs	100 inputs	200 inputs
0/100%	0.2%	0.1%	0.0%
5/95%	**85.4%**	84.8%	87.1%
10/90%	83.0%	84.6%	92.8%
15/85%	81.2%	88.8%	94.4%
20/80%	82.1%	90.3%	95.1%
25/75%	81.6%	91.5%	96.6%
30/70%	82.7%	**91.7%**	96.6%
35/65%	82.8%	91.2%	**96.9%**
40/60%	80.0%	90.8%	96.3%
45/55%	77.3%	90.2%	95.6%
50/50%	79.7%	89.2%	95.4%
55/45%	75.9%	88.8%	95.3%
60/40%	74.9%	87.2%	93.1%
65/35%	75.2%	86.5%	92.4%
70/30%	71.1%	84.7%	91.1%
75/25%	70.3%	85.7%	90.9%
80/20%	41.8%	83.6%	89.6%
85/15%	17.4%	80.9%	89.5%
90/10%	36.5%	30.2%	87.5%
95/5%	21.7%	57.2%	78.5%
100/0%	0.3%	0.6%	0.6%
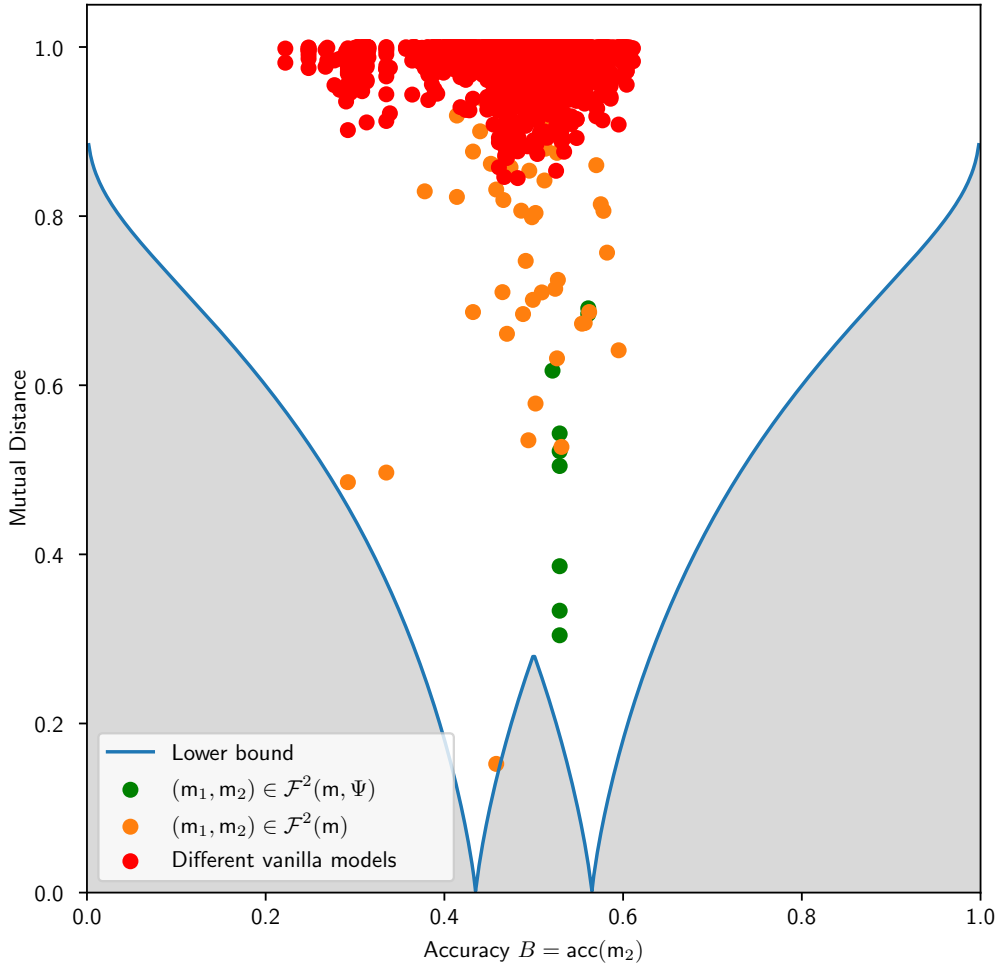
Figure 13: Comparison of the theoretical lower bound and the measured distance between models m_1 and m_2, when $A = \mathsf{acc}(m_1) = 0.45$ and $(m_1, m_2) \in \mathcal{F}^2(m)$ (orange), $(m_1, m_2) \in \mathcal{F}^2(m, \Psi)$ (green), or m_1 and m_2 are variants of different vanilla models (red).
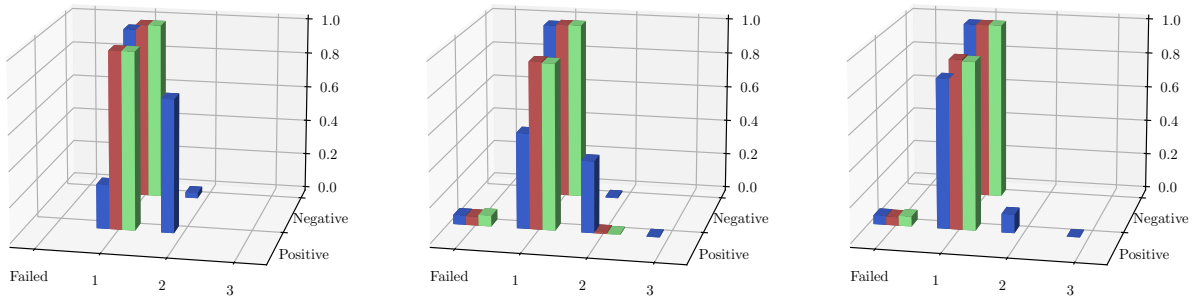

Figure 14: Histogram of the number of queries under $(\mathsf{D}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$ using worst-case score (8) when black-box returns top-1 (*blue*), top-3 (*red*) or top-5 (*green*) decisions. Family considered from left to right: $\mathcal{F}(m)$, $\mathcal{F}(m, \Psi)$, and $\mathcal{F}(m, \{\theta\})$.
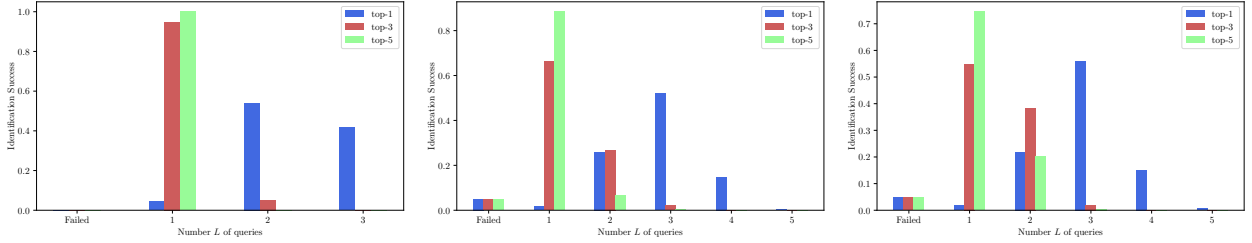
Figure 15: Histogram of the number of queries under $(\mathsf{I}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$ using worst-case score (8) when black-box returns top-1 (*blue*), top-3 (*red*) or top-5 (*green*) decisions. Family considered from left to right: $\mathcal{F}(\mathsf{m})$, $\mathcal{F}(\mathsf{m}, \Psi)$, and $\mathcal{F}(\mathsf{m}, \{\theta\})$.

Table 11: True Positive Rate per variation families under $(\mathsf{D}, \mathcal{F}(\mathsf{m}), \mathcal{A} \subsetneq \mathcal{B}, 1)$. False Positive Rate set to 5% and $L = 100$ queries submitted.

| Method | Param. | Finetune | | Half | Histo. | JPEG | Poster. | Prune | | | Random. |
		All	Last	Prec.				All	Filter	Last	Smooth.
IP-Guard [2]	BP [35] 50 iter.	0.5	92.3	**100**	27.3	**100**	9.2	72.7	89.2	**100**	26.1
FBI	Random	85.3	91.4	**100**	77.4	92.7	88.0	69.5	91.3	90.7	48.7
	30/70	94.1	**97.1**	**100**	90.3	94.7	95.4	89.5	**100**	95.8	79.1
	Entropy	**94.1**	**97.1**	**100**	**100**	**98.8**	**99.4**	**93.7**	**100**	**100**	**85.6**

28