DOI: 10.15514/ISPRAS-2025-37(6)-31

ExpressPrint: An Approach to Watermarking of Visual Foundation Models

[1,2] *A.S. Chistyakova, ORCID: 0000-0003-4896-4418 <a.chistyakova@ispras.ru>*
[1,3] *M.A. Pautov, ORCID: 0000-0003-0438-6361 <pautov@airi.net>*

[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

[2] *Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia.*

[3] *AIRI, 4th floor, building 2, 6, Presnenskaya embankment, Business complex "Empire", Moscow, 123112, Russia.*

Abstract. The substantial cost of training from scratch of visual foundation models (VFMs) on large and vast datasets motivates the models' owners to protect their intellectual property via ownership verification methods. In this work, we propose ExpressPrint, a novel approach to watermarking VFMs based on the fine-tuning of expressive layers of VFMs together with a small encoder-decoder network to embed the digital watermarks into a set of input images. Our method implies a small modification of expressive layers together with training an encoder-decoder neural network to extract user-specific binary messages from the hidden representations of certain input images. This method allows distinguishing between the foundation model provided to a user and independent models, thereby preventing unauthorized use of the model by third parties. We discover that the ability to correctly extract encoded binary messages from images transfers from a watermarked VFM to its functional copies obtained via pruning and fine tuning; at the same time, we experimentally show that non-watermarked VFMs do not share this property.

Keywords: visual foundation models; neural network watermarking; expressive layers; massive activations; trustworthy artificial intelligence.

For citation: Chistyakova A.S., Pautov M.A. ExpressPrint: An Approach to Watermarking of Visual Foundation Models. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 6, part 2, 2025, pp. 223-236. DOI: 10.15514/ISPRAS-2025-37(6)-31.

ExpressPrint: метод создания цифровых водяных знаков для визуальных базовых моделей

[1,2] *А.С. Чистякова, ORCID: 0000-0003-4896-4418 <a.chistyakova@ispras.ru>*
[1,3] *М.А. Паутов, ORCID: 0000-0003-0438-6361 <pautov@airi.net>*

[1] *Институт системного программирования им. В.П. Иванникова РАН, Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*

[2] *Московский государственный университет имени М.В. Ломоносова, Россия, 119991, Москва, Ленинские горы, д. 1.*

[3] *AIRI, Россия, 123112, Москва, Деловой комплекс «Империя», Пресненская набережная, д. 6, стр. 2, 4 этаж.*

Аннотация. Значительные затраты на обучение визуальных базовых моделей с нуля на больших и обширных наборах тренировочных данных мотивируют владельцев моделей прибегать к использованию методов защиты интеллектуальной собственности. В данной работе предложен метод ExpressPrint – новый подход к созданию цифровых водяных знаков для визуальных базовых моделей, основанный на дообучении наиболее выразительных слоев модели совместно с небольшой нейронной сетью типа “кодировщик-декодировщик” с целью встраивания цифровых водяных знаков в отложенный набор входных изображений. Предложенный метод подразумевает незначительные модификации выразительных слоев модели наряду с обучением нейронной сети типа “кодировщик-декодировщик” для извлечения специфичных для пользователя бинарных сообщений из скрытых представлений входных изображений. Данный подход позволяет отличать модель, предоставленную в пользование по лицензии, от других версии модели, и, таким образом, предотвращать несанкционированное использование модели третьими лицами. В работе было обнаружено, что способность корректно извлекать закодированные бинарные сообщения из изображений передается от исходной базовой модели, к ее функциональным копиям, полученным посредством дообучения и прунинга; помимо этого показано, что независимые визуальные базовые модели, не подвергавшиеся нанесению цифровых водяных знаков, не обладают данным свойством.

Ключевые слова: визуальные базовые модели; цифровые водяные знаки для нейронной сети; выразительные слои; массивные активации; доверенный искусственный интеллект.

Для цитирования: Чистякова А. С., Паутов М. А. ExpressPrint: метод создания цифровых водяных знаков для визуальных базовых моделей. Труды ИСП РАН, том 37, вып. 6, часть 2, 2025 г., стр. 223–236 (на английском языке). DOI: 10.15514/ISPRAS–2025–37(6)–31.

1. Introduction

Although foundation models are useful tools that are deployed in a variety of practical scenarios from the fields of natural language processing [1], computer vision [2], biology [3], and many others [4], their training is costly in terms of both time and money, making the models valuable assets of their owners. Nowadays, user access to foundation models is mainly organized via subscription to a service where the model is deployed or via purchasing the license to use the specific instance of the model. Unfortunately, some users are violating the terms of use (for example, by integrating their instances of models into their services to make a profit or impermissibly distributing copies of the model). Consequently, it is reasonable that the models’ owners are willing to defend their intellectual property from unauthorized usage by third parties.

One of the prominent approaches to protecting the intellectual property rights (IPRs) of models is watermarking [5-7], a family of methods to embed specific information into the source model by modification of the latter. Ownership verification is then done by checking the existence of this information in the suspicious model. The other set of methods to protect IRPs is based on fingerprinting, which usually does not introduce any changes to the defending model [8-11]. Instead, these methods create a unique feature, or the fingerprint, of the source model; ownership verification

in this case is done by comparing the fingerprints of the defended model and the one extracted from the suspicious model.

In this work, we propose a method to watermark visual foundation models by embedding digital watermarks into hidden representations of certain input images. To choose a proper hidden representation to embed a watermark into, we utilize the concept of massive activations [12]: some blocks of a VFM tend to produce high-magnitude activations in response to various inputs; these activations usually dominate the ones of the subsequent layers. Here and below, we refer to the block (or layer) that produces massive activations to the expressive block (or layer). In our method, we experimentally verify that embedding a watermark into the representation of the expressive block allows us to protect the ownership of VFMs fine-tuned for different practical tasks, such as image classification and segmentation.

Our contributions are summarised as follows:

- We introduce ExpressPrint, a novel approach to watermarking visual foundation models. The proposed method is based on embedding digital watermarks into a hidden representation of a private set of input images of the VFM, where the choice of the proper representation is done by utilizing the concept of massive activations.
- We experimentally show that the proposed method allows us to distinguish between the watermarked VFM and the independent ones under the fine-tuning of the model for different practical tasks, such as classification and segmentation.
- We demonstrate that different VFM architectures can be watermarked by our method, showing the practical applicability of ExpressPrint. This work is the first, to the best of our knowledge, that addresses the problem of watermarking of visual foundation models.

2. Related Work

2.1 Foundation Models and Massive Activations

Visual foundation models, especially those based on Vision Transformers (ViT) [13], have become a dominant paradigm in modern computer vision due to their scalability and transferability across tasks. The development of Self-Supervised Learning (SSL) [14] methods in computer vision has led to the era of universal models, such as SimCLR [15], DINO [16], CLIP [17], and DINOv2 [18], that learn representations from unlabeled images and show impressive flexibility in solving diverse tasks with minimal labeled data for fine-tuning. However, the internal mechanisms, particularly the nature of neural activations, have so far remained little studied.

One emerging concept in the analysis of these models is massive activations [12] – unusually high responses in specific layers or tokens that play a significant role in decision-making. These activations tend to appear across various layers, often have consistently high magnitudes, and are frequently located at the same spatial or token positions across diverse input samples.

2.2 Protecting Intellectual Property via Watermarking and Fingerprinting

The application of watermarking and fingerprinting techniques to protect the intellectual property rights of neural networks has recently become an important topic in trustworthy artificial intelligence. In [19], the authors apply instruction tuning to fingerprint large language models: a predefined private key triggers a model to produce a specific text when present in the input prompt. The authors of [20] formalize the definition of artifact and fingerprint in large generative models based on the geometric properties of the training data manifold. In [8], authors propose to utilize artificially generated images in the attribution of image classifiers under model extraction attacks. In [21], authors discover that it is possible to reliably detect was trained on a synthetic output of a watermarked large language model, which discloses a potential privacy concern of neural network watermarking.

3. Problem Statement

In this work, we present a method to watermark visual foundation models by training an auxiliary network that embeds binary messages into hidden representations of input images of the source VFM. We start by introducing the notations used throughout the paper. Namely, let s be the dimension of an image, Ω be the space of visual foundation models, $f: R^s \to R^d$ be the source VFM that maps input images to embeddings of dimension d, h be the dimension of hidden image representation, and let $m \in \{0,1\}^k$ be the binary vector of length k. Let f be the composition $f(x) \equiv q(p(x))$, where $p: R^s \to R^h$ and $q: R^h \to R^d$ represent mappings from an image to the hidden representation and from the hidden representation to the output embeddings, respectively. In our method, we train two auxiliary models, namely, encoder $e: R^h \times \{0,1\}^k \to R^h$ that embeds the binary message m into the hidden representation $p(x)$ and decoder $d: R^d \to \{0,1\}^k$ that extracts binary messages from $q(x)$. In addition, we fine-tune the latter part of the source model, namely, q. Given the image x, the message m embedded into $p(x)$ and transform $\pi: \Omega \to \Omega$ that maps the foundation model to its functional copy, the goal of the method is two-fold: on the one hand, the decoder d should extract close messages from hidden representations of f and $\pi(f)$; on the other hand, given the model g which is functionally independent of f, the messages extracted from hidden representations of f and g should be far apart.

4. Proposed Method

We introduce ExpressPrint, a novel watermarking method designed to verify ownership of VFMs. ExpressPrint embeds user-specific binary signatures into internal feature representations of VFMs through fine-tuning a small number of expressive layers, accompanied by the joint training of lightweight encoder and decoder networks. This approach enables ownership verification by extracting digital fingerprints directly from model activations when provided with specific input images.

Unlike traditional watermarking techniques that modify model weights or outputs, our method introduces minimal architectural changes while preserving the functional capacity of the model.

4.1 Watermarking Pipeline

The watermarking pipeline is illustrated in Fig. 1. The process consists of the following steps:

1) Embedding of the watermark. Given input image x and user-specific binary message m, we use a lightweight encoder network that injects m into a selected channel of the internal activation of the predefined expressive block. This injection is performed via a forward hook attached to an expressive transformer block.
2) Propagation and Decoding. The modified representation of x is propagated through the remaining trainable layers of the VFM. At a later block, another hook triggers a decoder network to extract the binary message m', which is then compared to m.
3) Training. The encoder, decoder, and a small set of trainable VFM layers are optimized jointly to minimize the decoding loss while maintaining task performance. The remaining VFM weights are frozen to prevent degradation.

4.2 Expressive Layer Selection

The core idea behind ExpressPrint is to embed binary signatures into expressive regions of a model's latent space. We build on the observation that massive activations tend to emerge in later blocks of VFMs and appear for the majority of the input images. Hence, we hypothesize that these high-activation regions are suitable for watermark embedding due to their huge impact on the image representations in the subsequent blocks of the model [12].

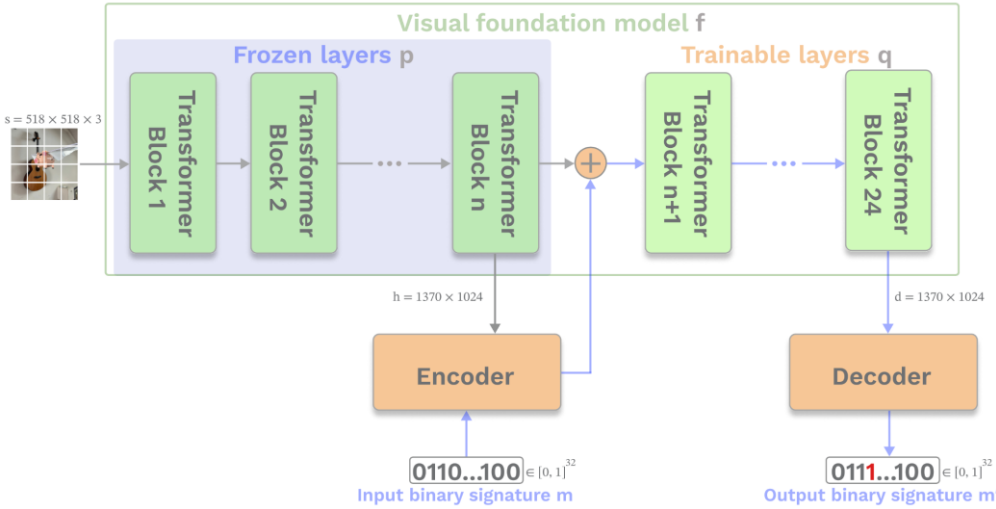

Fig. 1. Schematic illustration of the proposed method. To embed a watermark, we use an auxiliary learnable encoder network and inject a user-specific binary signature into a selected channel of the internal activation. To extract the watermark, we apply an auxiliary decoder network that extracts a binary message from the image representation in a later transformer block.

To identify such regions, we analyze the activation patterns across the blocks of a pre-trained VFM. Specifically, we pass 100 randomly selected natural images through the model and compute, for each block, the average of the top-5 absolute activation values per image. These per-block averages are then aggregated over all images to yield a global activation profile. As shown in Fig. 2, we observe an explosion of activations in the final blocks of the model architecture. This motivates our selection of these blocks as carriers for signature embedding. For each selected expressive block, we further localize the most influential tokens – the ones that are most critical for the block's output. For each token, we compute the absolute values of output activations and choose the token as an outlier if the corresponding output activation increases drastically after some block (namely, if on a particular layer its z-score increases up to 100). We propose to use these outlier tokens as internal activation anchors for binary message injection.

4.3 Loss Function

Our training objective is the combination of two terms: given an input sample, the first one ensures that the feature representations of the watermarked and original models do not deviate much; the second term forces the extracted binary message to be close to the embedded one. Specifically, given $q = q(x)$ as the representation of the expressive part of the source model and $q' = q'(x)$ as the representation of the expressive part of the watermarked model, the objective function is presented in the form below:

$$L(q, q', x, m, m') = L_{fp}(q, q', x) + \lambda L_{sig}(m, m'), \quad (1)$$

where:

- $L_{fp}(q, q', x) = ||q(x) - q'(x)||_2$ is the feature preservation loss (namely, mean squared error between the original representation and the modified representation);
- $L_{sig}(m, m') = ||m - m'||_2$ is the soft distance between the extracted and ground-truth signatures (namely, mean squared error between the extracted and ground-truth signatures);
- λ is a scalar parameter that controls the trade-off between feature fidelity and signature reconstruction.

This formulation ensures that embedded watermarks are extracted from the fingerprinted models while minimizing the impact on the feature representation.

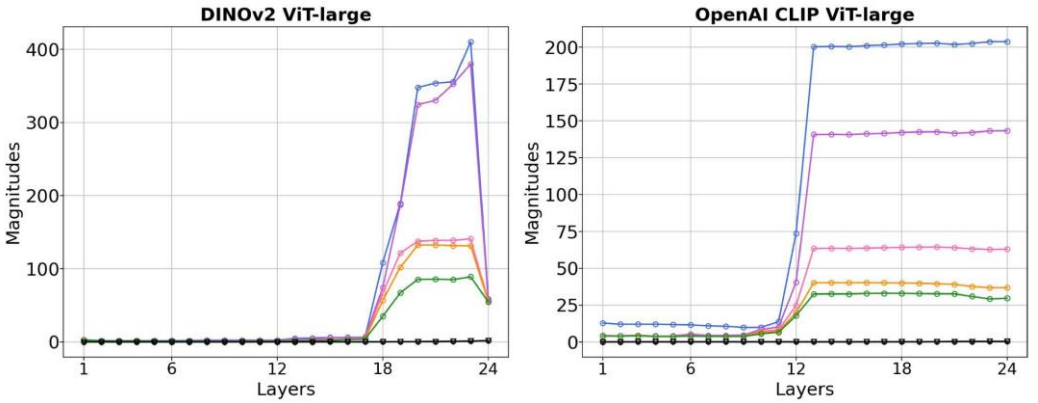

Fig. 2. Top-5 activation values across layers. It is noteworthy that starting from a particular layer, the magnitudes of activations increase drastically. In our work, such a layer represents the beginning of the expressive part of a visual foundation model.

4.4 Assessing the Performance of the Watermarking Method

To evaluate the performance of the proposed method, given a specific user message m and input image x, we compute the distance ρ between the extracted binary message $m'(f,x)$ and m. Note that we specifically indicate that the extracted message depends both on the input image and the model from which it is extracted. We measure the distance as the number of bits which differ in m and $m'(f,x)$:

$$\rho(m,m'(f,x)) \;=\; \sum_{i=1}^{n} \quad 1[m_i(f,x) \neq m'_i], \tag{2}$$

where 1 is the indicator function.

Recall that a good watermarking method has to satisfy two conditions: on the one hand, given an input image x, for the watermarked model f, the distance has to be close to 0; on the other hand, for a separate (independent) model, the distance has to be close to n.

In this work, the decision rule that is used to evaluate whether the given network is watermarked is the comparison of the distance with a predefined threshold: given a single input image x, we treat f as the watermarked iff

$$\rho(m,m'(f,x)) \;<=\; t, \tag{3}$$

where $t >= 0$ is the threshold value. The case of many samples is discussed in Section 5.3.

4.4.1 Setting the Threshold Value

We set the threshold by formulating a hypothesis test: the null hypothesis, $H_0 =$ "$the\ model\ f\ is\ not\ watermarked$" is tested against an alternative hypothesis, $H_1 =$ "$the\ model\ f\ is\ watermarked$", for a given model f. In this work, we assume that the messages $m'(g,x)$ extracted from all the non-watermarked models g are distributed uniformly over all bit strings of length n. Having said so, we estimate the probability of false acceptance of hypothesis H_1 (namely, FPR_1) as follows:

$$FPR_1 = P_{m'(g,x)}(\rho(m,m'(g,x)) \;<=\; t) \;=\; \sum_{q\in[0,\dots,t]} \quad {C_n}^q (1-p)^q p,^{n-q} \tag{4}$$

where $p = P_{m\prime(g,x)}(m_i = m'(g,x)_i)$ is the probability that i'th bits in m and $m'(g,x)$ coincide. In our work, we experimentally evaluate that p is close to 0.5 for all the indices i. To choose a proper threshold value for t, we set up an upper bound for FPR_1 as ε and solve for t, namely,

$$t = arg\ \max_{t'<k}(\sum_{q\in[0,\dots,t']} C_n{}^q(1-p)^q p^{n-q}), subject\ to\ \sum_{q\in[0,\dots,t']} C_n{}^q(1-p)^q p^{n-q} < \varepsilon \quad (5)$$

For example, if $n = 32$ and $\varepsilon = 10^{-5}$, then $t = 4$.

5. Experiments

5.1 Models and Datasets

We conducted our experiments using two large-scale VFMs: CLIP [17] and DINOv2 [18]. For evaluation, we used a subset of images from the ImageNet dataset [22]. To train models on downstream tasks (namely, for classification and segmentation), we utilized three domain-specific datasets:

- E-commerce Product Images [23]: This dataset consists of 18,175 product images categorized into 9 major classes based on Amazon's product taxonomy. It is primarily used for image-based product categorization.
- Oxford-IIIT Pet Dataset [24]: A classification and segmentation dataset containing 37 pet categories (dogs and cats), with approximately 200 images per class. It includes both breed labels and foreground-background segmentation masks.
- FoodSeg103 [25]: A food image segmentation dataset containing 7,118 images annotated with fine-grained pixel-wise labels for over 100 food categories. It supports both semantic segmentation and instance-level analysis of food items.

5.2 Training Details

5.2.1 Watermark Injection

We initialized both VFMs with publicly available pre-trained weights [27-28]. To embed and extract binary watermarks within their internal representations, we designed a lightweight encoder-decoder architecture. For each image, we randomly and uniformly sampled a 32-bit binary vector and assigned it as the corresponding watermark for this image.

The watermark encoder consisted of two fully connected layers and processed the concatenation of a selected feature channel with the binary watermark vector. Namely, the encoder maps the pair (internal image representation, binary message) to a vector from the dimension of the hidden representation:

$$e(p(x), m) = u \in R^h. \quad (6)$$

This encoded perturbation was injected into a specific transformer block and channel via a forward hook.

Similarly, the watermark decoder, also composed of two fully connected layers, was used to reconstruct the watermark from the modified features extracted at a later block. During inference, outputs were thresholded at 0.5 to obtain binary predictions:

$$d(q(e(p(x), m)))) = v \in R^k;\ m'_i = 1(v_i > 0.5)\ \forall\ i \in [1, \dots, k]. \quad (7)$$

Based on prior activation analysis (see Fig. 2), we chose block 12 in CLIP ViT-L/14 and block 18 in DINOv2 ViT-L/14 for the embedding of the watermarks (note that in the corresponding blocks, massive activations first emerge). All transformer layers prior to the embedding point were frozen during training, while the encoder, decoder, and all subsequent blocks are subject to change during watermark embedding. Further discussion of this selection is provided in Section 5.3.

During training, we used AdamW optimizer with learning rates of 10^{-4} for the VFM backbones and 10^{-5} for watermark modules (namely, e and d). Both models were trained for 5 epochs with a batch size of 16.

5.2.2 Fine-tuning for Downstream Tasks

We used a full fine-tuning strategy in all experiments to preserve watermark robustness against representation shifts introduced by fine-tuning. Experiments were conducted on both classification and segmentation tasks. For the classification tasks, three learning rate schedulers were evaluated: constant (no scheduler), cosine annealing, and linear decay; for the segmentation tasks, no learning rate schedulers were used. The training was performed using the AdamW optimizer for 10 epochs.

5.2.3 Pruning

To investigate the impact of model sparsity on both classification accuracy and watermark robustness, we applied post-training unstructured L1-norm pruning to the entire model. We evaluated two sparsity levels: moderate pruning, where 20% of the lowest-magnitude weights were zeroed out, and aggressive pruning, where 40% of the weights were removed. This procedure enabled us to assess the effect of varying sparsity levels on watermark reconstruction. Note that the unrestricted L1-norm pruning is used purely as the baseline to illustrate the robustness of the proposed method to a model's modification.

5.3 Signature Location Selection

Given the source model f, the set of input samples $\{x_1, x_2, \dots, x_N\}$, false positive rate threshold from Eq. (4). and corresponding distance threshold t from Eq. (5), we compute the watermark detection rate

$$r = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\rho(m(x_i), m'(f, x_i)) <= t], \quad (8)$$

where $N = 1000$. Note that here we explicitly write $m(x_i)$ to indicate that for different input images, watermark messages may differ.

For the length of the watermark $k = 32$ and two similarity thresholds (namely, $t = 0$ and $t = 4$), we study the dependence of the watermark detection rate on the number n of the last frozen layer (see Fig. 1 for clearance). Note that the value of the watermark detection rate at $t = 0$ indicates the ability of the method to extract the same watermarks that were embedded. To evaluate the robustness of the embedded watermarks to downstream task adaptation, we fine-tuned the CLIP model on a semantic segmentation task (see Section 5.2 for details). The values of the hyperparameters were chosen to satisfy the trade-off between the feasibility of the watermark embedding and its robustness under the perturbations of the watermarked model.

In Table 1, we report the watermark detection rates and average bitwise distance between an embedded watermark and an extracted binary message (see Eq. (2)).

The results indicate that early transformer blocks have low watermark detection rates and high reconstruction errors, making them unsuitable for embedding. In contrast, blocks 12, 13, 15, 21, and 22 show high detection rates and low errors, demonstrating better stability and reliability for watermark embedding. Block 12, in particular, provides the best balance between detection accuracy and reconstruction quality. Notably, this is also the first layer where massive activations begin to emerge, which may contribute to its effectiveness as an embedding point.

While watermarks embedded closer to the end of the transformer (e.g., in blocks 21 and 22) are extracted with reasonably high accuracy, their robustness degrades after fine-tuning. This is likely because the later layers are more heavily modified during task-specific adaptation, which negatively impacts the detection rate. Therefore, mid-level blocks such as block 12 offer a more reliable trade-off between initial detection performance and robustness to downstream fine-tuning.

Table 1. Dependency of the watermark detection rate on the number of the first expressive transformer block, n. The architecture of the source visual foundation model is CLIP.

Block number, n	Watermark detection rate, $r \uparrow$ $t = 0$	Average bitwise error $\downarrow$	Watermark detection rate after fine-tuning, $r \uparrow$ $t = 4$
1	0.000	15.959	0.000
2	0.000	16.033	0.000
10	0.000	14.382	0.000
11	0.000	11.255	0.000
12	0.938	**0.143**	**0.983**
13	**0.960**	0.607	0.980
14	0.483	1.766	0.610
15	0.940	0.885	0.810
21	0.939	0.328	0.945
22	0.931	0.150	0.864
23	0.569	0.852	0.882
24	0.000	15.905	0.000

6. Results

In this section, we provide the results of experiments and elaborate on them.

6.1 Overall Results

In Fig. 3, we report the dependency of the watermark detection rate on the threshold value of the false positive rate from Eq. (4). We evaluated classification and segmentation tasks and studied the effect of unstructured pruning. Details of experiments are provided in Section 5.2.

6.2 Comparison with Fingerprinting Methods

To assess the effectiveness of the proposed method, we compare it against the state-of-the-art fingerprinting method, ADV-TRA [26]. The main idea of this method goes as follows. Given the input sample x of the ground truth class y, an index of the target class y_t, and the source model f, the set of adversarial examples $T = (x, x_1, x_2, \dots, x_l)$ is computed such that

$$x_{i+1} = x_i - s_i \, sign(\nabla L(f, x_{i,}, y_t)),$$

where L is the loss function guiding the optimization towards y_t, and s_i is the scalar variable denoting the step size on i'th iteration. Then, this set T is used to compare the predictions of the source model f and a suspect model, f_{sus}, to decide whether f_{sus} is a stolen copy of f. It is worth mentioning that the process of generation of the model's fingerprint, namely, the predictions on the adversarial examples, is downstream task-dependent: the owner of the model has to use an auxiliary classification head to compute adversarial examples. This limitation makes ADV-TRA barely feasible for other downstream tasks.

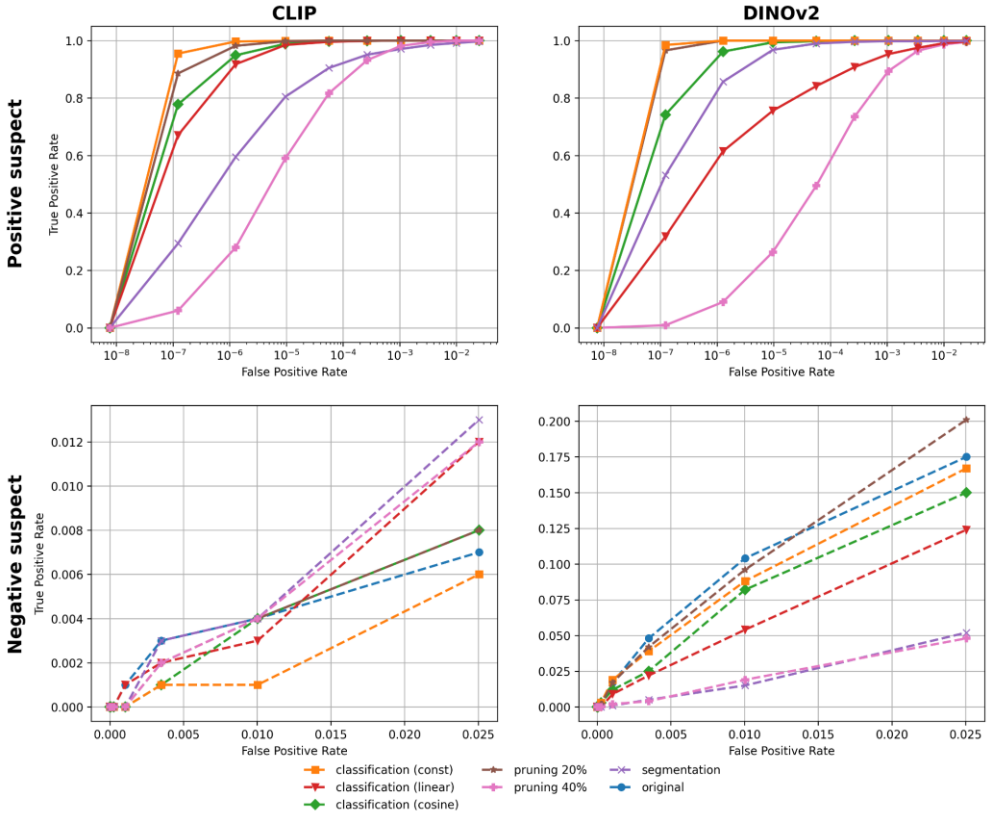

Fig. 3. Effectiveness of ExpressPrint in ownership verification problem. We consider several downstream tasks (classification and segmentation), and VFM architectures (CLIP and DINOv2) and evaluate our approach against fine-tuning and pruning of the watermarked model.

To compare ExpressPrint with ADV-TRA on different downstream tasks and assess their robustness under different types of model perturbations, we compute the watermark detection rate (or fingerprint detection rate, in the case of ADV-TRA). Note that the higher the detection rate for positive suspect models, the more precisely a method detects functionally stolen models; on the other hand, the lower the detection rate in the case of the negative suspect models, the less frequently a method detects an independent model as the functional copy of the source one. In Table 2, we report the quantitative results of the comparison.

There are two types of suspect models in Table 2: positive suspect models and negative suspect models. While positive suspect models represent the set of VFMs that are functionally connected to the source (watermarked) model, the negative suspect models are independent of the source VFM. The description of positive suspect models is presented in Sections 5.2.2. and 5.2.3; as the negative suspect models, we use different visual foundation models, namely, CLIP [17] and DINOv2 with

registers [29]. Note that the difference between DINOv2 and DINOv2 with registers is that the latter is equipped with additional learnable tokens which are shared among different input samples during the model's training.

*Table 2. Comparison of different fingerprinting techniques, the architecture of the source visual foundation model is DINOv2. We report a watermark (fingerprint) detection rate for both positive suspect models and negative suspect models. The best results are highlighted in **bold**. It is noteworthy that ExpressPrint outperforms ADV-TRA both in terms of true positive detection rate and false positive detection rate.*

Model Type		ADV-TRA	ExpressPrint
Positive suspect	classification (const)	0.703	**1.000**
	classification (linear)	0.012	**0.842**
	classification (cosine)	0.123	**0.998**
	segmentation	0.000	**0.990**
	pruning 20%	**1.000**	**1.000**
	pruning 40%	0.086	**0.495**
Negative Suspect	different architecture (DINOv2 with registers)	0.012	**0.000**
	different architecture (CLIP)	**0.000**	**0.000**

In Table 3, we report the computation overhead of the methods. It is worth mentioning that ADV-TRA is excessively time-consuming in terms of fingerprint generation. At the same time, signature verification times (or watermark extraction times for ExpressPrint) do not differ much for these two approaches. Note that the first two columns of the Table 3 (namely, Signature injection (min) and Signature generation (min)) indicate the time required for watermark (or fingerprint) preparation; although for a given model this procedure has to be conducted only once, the faster it is, the more applicable the corresponding method for the large-scale VFMs.

7. Discussion and Limitations

We experimentally show that ExpressPrint allows us to distinguish between independent models and functional copies of the given watermarked visual foundation model. It is worth mentioning that ExpressPrint is a downstream task-agnostic approach: the model's owner has to prepare a set of input images and perform the watermark embedding procedure only once for a given instance of the model; then, the watermarked model remains detectable by our method after fine-tuning to a particular downstream task (for example, image classification and segmentation). In comparison to ExpressPrint, ADV-TRA, the state-of-the-art fingerprinting approach does not provide a satisfactory level of fingerprint detection rate when the fingerprinted model is fine-tuned for a different downstream task (namely, segmentation; see Table 2). More than that, the proposed method is robust to the pruning of the protected model and yields very low false detection rates (namely, zero for the experiments from Table 2).

Table 3. Comparison of computational overheads for ADV-TRA and ExpressPrint, the architecture of the source visual foundation model is DINOv2. We report cumulative time in minutes for N=1000 images.

	Signature injection (min)	Signature generation (min)	Signature verification (min)
ADV-TRA	0.000	1663.70	3.412
ExpressPrint	31.533	3.105	4.017

Among the limitations of ExpressPrint, we indicate the necessity to have white-box access to the first expressive layer of a suspect model: to verify that a model allows to encrypt and decrypt a certain message, one has to pass the image and corresponding binary string to it (see Eq. (7)).

8. Conclusion and Future Work

In this work, we propose ExpressPrint, a novel watermarking approach for visual foundation models. This method is model agnostic and allows to protect a VFM in a downstream task-independent manner: our experiments show that the proposed method allows to reliably detect the functional copies of a particular foundation model obtained by the fine-tuning and pruning both for image classification and segmentation. On the other hand, we verify that ExpressPrint does not detect benign, independent models as functional copies of the watermarked VFM, which makes the method applicable in practical scenarios. We compared the effectiveness and computation overhead of the proposed method with the state-of-the-art fingerprinting approach, ADV-TRA, and showed that the watermark detection rate and false positive detection rate of ExpressPrint are superior to the ones of ADV-TRA; at the same time, the proposed approach is significantly faster. Important directions of future work include an adaptation of the ExpressPrint to allow black-box inference of the suspect models.

References

[1]. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[2]. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125, 1(2), 3.

[3]. Ma, Jun, and Bo Wang. "Towards foundation models of biological image segmentation." Nature Methods 20.7 (2023): 953-955.

[4]. Schneider, Johannes, Christian Meske, and Pauline Kuss. "Foundation models: a new paradigm for artificial intelligence." Business & Information Systems Engineering 66.2 (2024): 221-231.

[5]. Uchida, Y., Nagai, Y., Sakazawa, S., & Satoh, S. I. (2017, June). Embedding watermarks into deep neural networks. In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (pp. 269-277).

[6]. Guo, J., & Potkonjak, M. (2018, November). Watermarking deep neural networks for embedded systems. In 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 1-8). IEEE.

[7]. Li, Y., Zhu, L., Jia, X., Jiang, Y., Xia, S. T., & Cao, X. (2022, June). Defending against model stealing via verifying embedded external features. In Proceedings of the AAAI conference on artificial intelligence (Vol. 36, No. 2, pp. 1464-1472).

[8]. Pautov, M., Bogdanov, N., Pyatkin, S., Rogov, O., & Oseledets, I. (2024, August). Probabilistically robust watermarking of neural networks. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (pp. 4778-4787).

[9]. Quan, Y., Teng, H., Xu, R., Huang, J., & Ji, H. (2023). Fingerprinting deep image restoration models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 13285-13295).

[10]. He, Z., Zhang, T., & Lee, R. (2019). Sensitive-sample fingerprinting of deep neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4729-4737).
[11]. Peng, Z., Li, S., Chen, G., Zhang, C., Zhu, H., & Xue, M. (2022). Fingerprinting deep neural networks globally via universal adversarial perturbations. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 13430-13439).
[12]. Sun M., Chen X., Kolter J. Z., and Liu Z. Massive Activations in Large Language Models. (2024) In First Conference on Language Modeling.
[13]. Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., and Gelly S. (2020) An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In International Conference on Learning Representations, 2020.
[14]. Balestriero R., Ibrahim M., Sobal V., Morcos A., Shekhar S., Goldstein T., Bordes F., Bardes A., Mialon G., Tian Y., Schwarzschild A., Wilson A. G., Geiping J., Garrido Q., Fernandez P., Bar A., Pirsiavash H., LeCun Y., and Goldblum M. (2023) A Cookbook of Self-Supervised Learning. arXiv:2304.12210.
[15]. Chen T., Kornblith S., Norouzi M., and Hinton G. (2020) A simple framework for contrastive learning of visual representations. In International conference on machine learning, PmLR, 2020, pp. 1597–1607.
[16]. Caron M., Touvron H., Misra I., Jégou H., Mairal J., Bojanowski P., and Joulin A. (2021) Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9650–9660.
[17]. Radford A., Kim J. W., Hallacy C., Ramesh A., Goh G., Agarwal S., Sastry G., Askell A., Mishkin P., and Clark J. (2021) Learning transferable visual models from natural language supervision. In International conference on machine learning, PmLR, 2021, pp. 8748–8763.
[18]. Oquab M., Darcet T., Moutakanni T., Vo H., Szafraniec M., Khalidov V., Fernandez P., Haziza D., Massa F., and El-Nouby A. (2024) DINOv2: Learning Robust Visual Features without Supervision, Trans. Mach. Learn. Res. J., 2024, pp. 1–31.
[19]. Xu, J., Wang, F., Ma, M., Koh, P. W., Xiao, C., & Chen, M. (2024, June). Instructional Fingerprinting of Large Language Models. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) (pp.-3277-3306).
[20]. Song, H. J., Khayatkhoei, M., & AbdAlmageed, W. (2024). Manifpt: Defining and analyzing fingerprints of generative models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10791-10801).
[21]. Sander, T., Fernandez, P., Durmus, A., Douze, M., & Furon, T. (2024). Watermarking makes language models radioactive. Advances in Neural Information Processing Systems, 37, 21079-21113.
[22]. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp.-248-255).
[23]. Ecommerce Product Images for Product Categorization - https://www.kaggle.com/datasets/fatihkgg/ecommerce-product-images-18k, accessed 13.05.2025.
[24]. The Oxford-IIIT Pet Dataset - https://www.kaggle.com/datasets/tanlikesmath/the-oxfordiiit-pet-dataset, accessed 13.05.2025.
[25]. FoodSeg103 - https://huggingface.co/datasets/EduardoPacheco/FoodSeg103, accessed 13.05.2025.
[26]. Xu, T., Wang, C., Liu, G., Yang, Y., Peng, K., & Liu, W. (2024). United We Stand, Divided We Fall: Fingerprinting Deep Neural Networks via Adversarial Trajectories. Advances in Neural Information Processing Systems, 37, 69299-69328.
[27]. CLIP model for timm - https://huggingface.co/timm/vit_large_patch14_clip_224.openai, accessed 16.05.2025.
[28]. DINOv2 model for timm - https://huggingface.co/timm/vit_large_patch14_dinov2.lvd142m, accessed 16.05.2025.
[29]. Darcet, T., Oquab, M., Mairal, J., & Bojanowski, P. Vision Transformers Need Registers. In The Twelfth International Conference on Learning Representations.

Информация об авторах / Information about authors

Анна Сергеевна ЧИСТЯКОВА – старший лаборант Центра доверенного искусственного интеллекта ИСП РАН. Научные интересы: исследование и разработка нейросетевых архитектур, устойчивых к состязательным атакам.

Anna Sergeevna CHISTYAKOVA – Senior Assistant at ISP RAS Research Center for Trusted Artificial Intelligence. Research interests: research and development of neural network architectures aimed at robustness to adversarial attacks.

Михаил Александрович ПАУТОВ – кандидат компьютерных наук, научный сотрудник AIRI, Центра доверенного искусственного интеллекта ИСП РАН. Сфера научных интересов: линейная алгебра, теория больших отклонений, доказуемая устойчивость нейронных сетей, цифровые водяные знаки.

Mikhail Aleksandrovich PAUTOV – Cand. Sci. (Phys.-Math.) in computer science, research scientist at AIRI and ISP RAS Research Center for Trusted Artificial Intelligence. His research interests include linear algebra, large deviations theory, certified robustness and digital watermarking.