Exploring Self-Supervised Vision Transformers for Deepfake Detection: A Comparative Analysis

Huy H. Nguyen[1], Junichi Yamagishi[1,2], and Isao Echizen[1,2,3]
[1]National Institute of Informatics, Japan [2]SOKENDAI, Japan [3]The University of Tokyo, Japan
{nhhuy, jyamagis, iechizen}@nii.ac.jp

Abstract

This paper investigates the effectiveness of self-supervised pre-trained transformers compared to supervised pre-trained transformers and conventional neural networks (ConvNets) for detecting various types of deepfakes. We focus on their potential for improved generalization, particularly when training data is limited. Despite the notable success of large vision-language models utilizing transformer architectures in various tasks, including zero-shot and few-shot learning, the deepfake detection community has still shown some reluctance to adopt pre-trained vision transformers (ViTs), especially large ones, as feature extractors. One concern is their perceived excessive capacity, which often demands extensive data, and the resulting suboptimal generalization when training or fine-tuning data is small or less diverse. This contrasts poorly with ConvNets, which have already established themselves as robust feature extractors. Additionally, training and optimizing transformers from scratch requires significant computational resources, making this accessible primarily to large companies and hindering broader investigation within the academic community. Recent advancements in using self-supervised learning (SSL) in transformers, such as DINO and its derivatives, have showcased significant adaptability across diverse vision tasks and possess explicit semantic segmentation capabilities. By leveraging DINO for deepfake detection with modest training data and implementing partial fine-tuning, we observe comparable adaptability to the task and the natural explainability of the detection result via the attention mechanism. Moreover, partial fine-tuning of transformers for deepfake detection offers a more resource-efficient alternative, requiring significantly fewer computational resources.

1. Introduction

In recent years, deepfake detection has emerged as a highly investigated field driven by the increasing prevalence of synthetic media [3]. Transfer learning, a commonly adopted strategy in computer vision, has also been widely used in deepfake detection [46, 37]. The selection of an appropriate backbone architecture plays an important role, serving not only as a feature extractor but also as a regularizer to prevent overfitting. Previous studies have predominantly relied on ConvNets that are pre-trained using supervised learning on ImageNet. However, with the recent advancements in transformer architectures [49], such as CLIP [35] and GPT-4 [1], particularly in multi-modal tasks, there has been growing interest in exploring their efficacy for deepfake detection. Despite their demonstrated success in various domains, the adoption of pre-trained ViTs [16] as feature extractors, especially large ones, has thus far been met with hesitation in the deepfake detection community. This reluctance stems from concerns about their immense capacity, which may exceed the requirements of the task and lead to potential overfitting, as well as their demanding resource requirements regarding training or fine-tuning data and computational resources. In contrast, ConvNets have already established themselves as robust feature extractors.

The advent of SSL has revolutionized the field of transformers, beginning with natural language processing (NLP) models such as BERT [25] and GPT [36]. Subsequently, DINO [5] showcased the successful adaptation of SSL on ViTs, resulting in robust feature extractors and enabling explicit semantic segmentation of images—an ability not readily available in supervised ViTs. The introduction of registers [13] in DINOv2 [34] further validated these capabilities, demonstrating their effectiveness in transfer learning across various downstream tasks. Specifically, in the realm of deepfake detection, the preliminary work of Cocchi *et al.* [11] demonstrated that detectors using either a basic k-NN classifier or a linear classifier equipped with pre-trained frozen DINO backbones can effectively identify images generated by Stable Diffusion models [38].

The current study substantially expands upon the findings of Cocchi *et al.* [11] in several key aspects. Our contributions can be summarized as follows:

- We conduct an extensive comparative study to explore the utilization of pre-trained vision transformers in deepfake detection from two perspectives: utilizing their frozen backbones as multi-level feature extractors, a method increasingly utilized in the literature, and partially fine-tuning their final transformer blocks.

- We highlight the advantages of partially fine-tuning the final blocks, demonstrating improvements in performance and natural explainability of the detection result via the attention mechanism, despite being fine-tuned on a small dataset with binary class annotations.

- We conclude that leveraging self-supervised learning on vision transformers, pre-trained using large datasets unrelated to deepfake detection, leads to a superior performance on the detection of various deepfake images and videos compared to utilizing supervised pre-training.

2. Related Work

2.1. Self-supervised vision transformers

Caron *et al.* [5] argued that image-level supervision often oversimplifies rich visual information, reducing it to a single concept from a predefined category set. To address this limitation, they applied SSL on ViTs [16], particularly DeiT [47], to improve feature representation, leading to the development of DINO. This approach can be seen as a type of knowledge distillation [21] without explicit labels, leveraging techniques such as a momentum encoder [19], multi-crop training [4], and the utilization of small patches with ViTs. Through SSL training, DINO showcased a remarkable performance across various tasks, including image classification, image retrieval, copy detection, semantic layout discovery in scenes, video instance segmentation, probing the self-attention map, and transfer learning via fine-tuning on downstream tasks. Importantly, SSL provides explicit information about the semantic segmentation of the image, a capability not clearly present in supervised ViTs.

The introduction of DINOv2 [34] primarily focused on accelerating and stabilizing training at scale using a significantly larger dataset comprising images sourced from curated and uncurated data sources. A subsequent work by Darcet *et al.* [13] identified and characterized artifacts in the feature maps of supervised and self-supervised ViTs. They proposed a straightforward yet effective solution that augments the input sequence of ViTs with additional tokens, referred to as registers. These registers are utilized during training but discarded during inference. These enhancements contribute to the robustness and efficiency of training self-supervised ViTs, facilitating their broader application in various computer vision tasks, including deepfake detection.

2.2. Transformers in deepfake detection

In deepfake detection, transformers are primarily utilized in two ways: as feature refiners following ConvNets or as replacements for ConvNets as the main feature extractors.

The use of transformers as feature refiners has gained popularity following the introduction of the transformer architecture. In this approach, a ConvNet or an ensemble of ConvNets, often pre-trained and sometimes partially fine-tuned later, serves as the primary feature extractor. Subsequently, a transformer, typically a shallow one with few blocks, is trained to further refine the extracted features. Khan *et al.* [26] utilized XceptionNet [9] for feature extraction, followed by 12 transformer blocks for feature refining. Similarly, Wang *et al.* [50] utilized EfficientNet-B4 [44] as a feature extractor and introduced a multi-scale transformer as one branch while utilizing a frequency filter as another branch for additional processing of the extracted features. Coccomini *et al.* [12] opted for a smaller version of the feature extractor, EfficientNet-B0. Wang *et al.* [51] utilized ConvNets for feature extraction, followed by processing with a newly proposed convolutional pooling transformer before classification. Lin *et al.* [30] utilized a ConvNet for pre-processing, followed by a two-stream ConvNet and a transformer-based module. Notably, Zhao *et al.* [53] utilized XceptionNet as a feature extractor and proposed the interpretable spatial-temporal ViT, which comprises a decomposed spatial-temporal self-attention and a self-subtract mechanism to capture spatial artifacts and temporal inconsistency.

The second way of using transformers as the main feature extractors involves proposing novel architectures or leveraging the pre-trained large ViTs. In addition to combining patch embedding with EfficientNet-B7's features for pre-processing, Heo *et al.* [20] applied a distillation method of DeiT [47] on a transformer for deepfake detection. Guan *et al.* [18] introduced a local sequence transformer that models temporal consistency on sequences of restricted spatial regions. Ojha *et al.* [33] implemented nearest neighbor and linear probing on the frozen supervised pre-trained CLIP's ViT [35] for deepfake detection. Similarly, Cocchi *et al.* [11] utilized the same classifiers as Ojha *et al.* but additionally evaluated the self-supervised pre-trained DINO and DINOv2 as feature extractors. Liu *et al.* [31] developed a forgery-aware adapter integrated into a frozen CLIP's ViT, adapting image features to discern and integrate local forgery traces within image and frequency domains. Motivated by the work of Ojha *et al.*, Cocchi *et al.*, and Liu *et al.* [33, 11, 31], we conduct a comparative analysis on the use of self-supervised ViTs for deepfake detection from two perspectives: 1) utilizing their frozen backbones

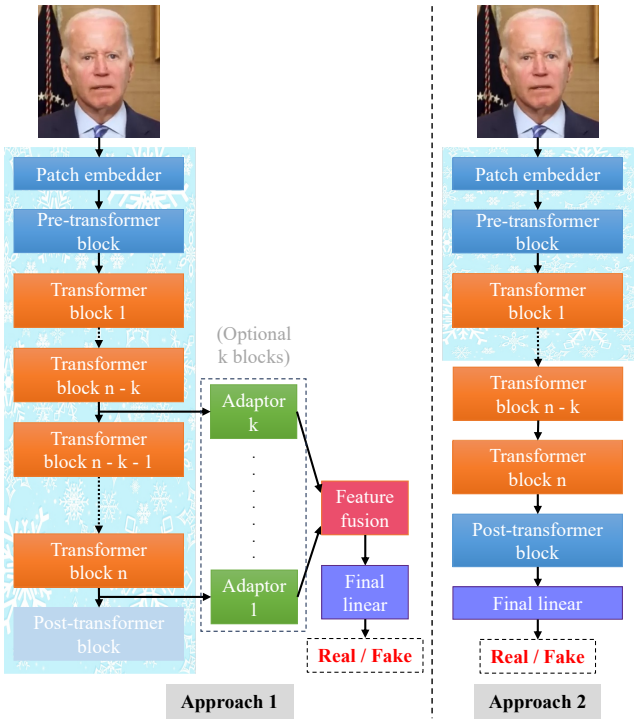

Figure 1. Overview of the two investigated approaches.

as multi-level feature extractors and 2) partially fine-tuning their final transformer blocks.

3. Methodology

In this section, we introduce two approaches that leverage pre-trained ViTs, as illustrated in Fig. 1. **Approach 1** generalizes previous methods in the literature (primarily applied to ConvNets) as a multi-level feature extractor on a frozen backbone. This approach has been widely adopted in the deepfake detection community [46, 37], often accompanied by sophisticated adaptors in recent work [31]. Notably, this strategy is endorsed as effective in the original DINO paper for various classification tasks [5], as well as in the detection of generative adversarial network (GAN) and diffusion images [33, 11]. **Approach 2** involves fine-tuning the final transformer blocks, a technique less favored due to the perception that transformers have large capacities and a high number of parameters.

3.1. Problem formalization

As a basic binary classification problem, given an input image I and a pre-trained backbone $\mathcal{B}$ with the classifier head removed, the objective is to construct a network $\mathcal{F}$ utilizing $\mathcal{B}$ to classify I as either "real" or "fake", corresponding to the binary labels 0 or 1. This can be represented as

$$\text{output} = \begin{cases} 1 & \text{if } \sigma(\mathcal{F}(\mathcal{B}(I))) \geq \tau \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function, mapping the output of $\mathcal{F}(\mathcal{B}(I))$ to a probability in the range $[0, 1]$, and τ is the threshold value. Note that the softmax function could be used instead of sigmoid to convert the logits extracted by $\mathcal{F}$ into probabilities, but utilizing softmax makes it easier to extend from binary to multi-class classification.

The backbone $\mathcal{B}$ begins with a pre-processing module and consists of n blocks. For simplicity, we denote the intermediate features of I extracted by block i as ϕ_i.

Regarding the value of τ, there are various methods for determining its optimal value, which can vary from one paper to another. In this work, we set τ to either 0.5 or to the threshold corresponding to the equal error rate (EER) calculated on the validation set, depending on the experiment settings.

3.2. Approach 1: Using frozen backbone as a multi-level feature extractor

In this approach, intermediate features ϕ_i can be further processed by an adaptor $\mathcal{A}$ (optional) before being fused with other intermediate features extracted by other blocks through a feature fusion operation Σ, followed by a classifier $\mathcal{C}$, typically a linear one. This approach is visualized on the left part of Fig. 1. The backbone $\mathcal{B}$ remains frozen. We utilize the k final intermediate features extracted by the k final blocks. This can be formalized by

$$\mathcal{F}(\mathcal{B}(I)) = \mathcal{C}\left(\sum_{i=n-k}^{n} \mathcal{A}_i(\phi_i)\right). \tag{2}$$

Due to the nature of the comparative study, we opted not to use complex architectures for the adaptor $\mathcal{A}$ and the classifier $\mathcal{C}$ here. Instead, we utilized dropout and linear layers for their construction. Further details can be found in Sec. 5.

3.3. Approach 2: Fine-tuning last transformer blocks

This approach is more straightforward than **Approach 1**. We append the new classifier $\mathcal{C}$ after the backbone $\mathcal{B}$, as visualized on the right part of Fig. 1. This can be formalized by

$$\mathcal{F}(\mathcal{B}(I)) = \mathcal{C}(\mathcal{B}(I)). \tag{3}$$

During fine-tuning, the first $n - k$ blocks are frozen. For the transformer backbones, the class (CLS) token and the register tokens (if they exist) are also unfrozen and fine-tuned along with the unfrozen k final blocks and the new classifier $\mathcal{C}$.

There are two major advantages of this approach compared to **Approach 1**:

- There are no additional parameters for the adaptors $\mathcal{A}$s and the feature fusion operation Σ. Given the already substantial size of recent feature extractors, par-

Table 1. Backbones used in the experiments.

Backbone	Architecture	Way of training	Dataset(s)	Images	Annotations
EfficientNetV2 Large [45]	ConvNet	Supervised	ImageNet-21K [14]	14M	Image classes
DeiT III L/16-LayerScale [48]	Transformer	Supervised	ImageNet-21K [14]	14M	Image classes
EVA-02-CLIP-L/14 [42]	Transformer	Supervised	LAION-2B [41] & COYO-700M [2]	2B	Image-text pairs
DINO (various versions) [5]	Transformer	Self-supervised	ImageNet-21K [14]	14M	Not used
DINOv2 (various versions) [34, 13]	Transformer	Self-supervised	LVD-142M [34]	142M	Not used

ticularly transformers, avoiding additional parameters is advantageous.

- *(For transformer backbones only)* Since the final transformer block and the tokens are fine-tuned, the attention weights to the CLS token are adapted to deepfake detection. They can be used naturally to visualize the focused area, similar to the visualization techniques used in the DINO papers [5, 34, 13]. This enhancement improves the detector's explainability, a crucial factor in deepfake detection.

4. Experimental Design

4.1. Backbones

We meticulously selected ConvNet and transformer backbones widely utilized in the domains of computer vision and forensics, prioritizing models with comparable sizes to ensure equitable and informative comparisons. Detailed specifications are provided in Table 1.

In terms of backbone architectures, we opted for EfficientNetV2 [45], given its reputation as a robust ConvNet architecture and the popularity of its first version in the forensics community. Additionally, we selected two well-known transformer-based models: DeiT III, the supervised-learning variant of DINO, and EVA-CLIP [42], an enhanced version of the renowned CLIP [35]. For supervised learning, we incorporated two prevalent annotations: the conventional class labels annotation and the multimodal image-text pairs annotation. We utilized the official pre-trained weights provided by the authors.

4.2. Datasets

We followed the data design of Nguyen *et al.* [32] by gathering a variety of images generated or manipulated by various deepfake methods to construct the main datasets. The details of the training, validation, and test sets are shown in Table 2. The datasets were designed to be balanced regarding the ratio of real and fake images and the number of images per training method, and were guaranteed not to overlap.

Real images were gathered from the VidTIMIT [40], VoxCeleb2 [10], FaceForensics++ (FF++) [39], Google DFD [17], Deepfake Detection Challenge Dataset (DFDC) [15], and Celeb-DF [29] datasets.

Table 2. Sizes of the main training, validation (val), and test (seen) sets, inspired by Nguyen *et al.*'s design [32], along with the sizes of the unseen validation and test sets from Ţânţaru *et al.*'s dataset [55].

Type	Real	Fake	Total
Training	44,037	55,963	100,000
Validation	13,200	13,000	26,200
Test	10,000	11,000	21,000
Validation (unseen)	1,900	10,700	12,600
Test (unseen)	900	3,600	4,500

One part of the **fake** images comprised images gathered from the FF++, Google DFD, Celeb-DF, DFDC, Deepfake-TIMIT (DF-TIMIT) [27], and YouTube-DF [28] datasets. The other part of the **fake** images were images generated by various GANs, including StarGAN [7], StarGAN-v2 [8], RelGAN [52], ProGAN [22], StyleGAN [23], and StyleGAN2 [24].

For **cross-dataset evaluation**, we used the dataset constructed by Ţânţaru *et al.* [55], which contains images generated or manipulated by diffusion-based methods. It is important to note that our training and validation sets (main dataset) above do not contain any diffusion images. The list of diffusion-based methods used here includes Perception Prioritized (P2) [6], Repaint-P2 [6, 55], Repaint-Latent Diffusion Model (LDM) [38, 55], Large Mask Inpainting (LaMa) [43], and Pluralistic [54].

Regarding the roles of the subsets, we used the training set for training or fine-tuning models and the validation sets for hyper-parameter selection, including the selection of the best checkpoints and determination of the EER thresholds, which were then used for testing. The test sets were used for evaluation and comparison.

4.3. Metrics

We used five metrics in our evaluation:

- Classification accuracy $= \frac{TP+\text{TN}}{\text{TP+TN+FP+FN}}$, where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively.
- True positive rate (TPR) $= \frac{TP}{TP+FN}$.

- True negative rate (TNR) = $\frac{TN}{TN+FP}$.
- Equal error rate (EER): value when false positive rate (FPR) equals false negative rate (FNR).
- Half total error rate (HTER): HTER = $\frac{\text{FPR}+\text{FNR}}{2}$.

5. Results and discussion

In this section, we first discuss **Approaches 1** and **2** in Sections 5.1 and 5.2, respectively. We compare the performances among different architectures and versions of DINO, as well as between DINOs and ConvNets and other transformers in more detail. Additionally, we implement improvements and conduct ablation studies on the best performing architectures to gain further enhancements and insights. Next, we evaluate selected models on the unseen test set in Sec. 5.3. Lastly, we visualize the attention maps of the fine-tuned DINO model and its original SSL pre-trained version on real and deepfake image examples in Sec. 5.4.

5.1. Approach 1: Using frozen backbone as a multi-level feature extractor

We initially validated the findings of Cocchi *et al.* [11] with some minor extensions by applying conventional classifiers to the features extracted by frozen SSL pre-trained DINO backbones (with small size (S) and base size (B)), specifically extracted from the CLS token. In addition to the nearest neighbors (k-NN) classifier and linear probing, we utilized k-means on the principal component analysis (PCA) features [32] and a two-layer perceptron classifier.

Regarding the PCA + k-means classifier, most models achieved accuracies of around 58%, which is slightly better than random guessing. As k-means is an unsupervised clustering method, this outcome suggests a partial ability of these pre-trained backbones to separate deepfakes without additional modules. For k-NN and linear probing, due to the complexity of the test set, the average accuracy was approximately 70%, which is lower than the results reported by Cocchi *et al.* on their diffusion dataset. The addition of an extra linear layer (the MLP) resulted in performance improvements ranging from about 2% to 14% in most cases, suggesting the potential for further enhancements.

Deepfake detection involves the identification of deepfake fingerprints, such as artifacts or irregular patterns. Therefore, relying solely on the CLS token may not be optimal. We therefore assessed the effectiveness of incorporating patch tokens and multiple intermediate features extracted from the final k blocks instead of solely from the final block. Additionally, we compared the performance of two feature fusion techniques: weighted sum (WS) and concatenation (concat). Since utilizing all tokens (CLS and patch tokens) results in huge feature sizes, we only evaluated the weighted sum technique in this case. Regarding the DINO backbones, in addition to small and base sizes, we also evaluated large (L) and giant (G) sizes, which are only publicly available in DINOv2. The results are presented in Table 4.

Table 3. Performances of four conventional classifiers on various DINO and DINOv2 architectures.

ViT backbone	DINO version	PCA + k-means	k-NN	Linear	MLP (2 layers)
S/8	1	58.69	69.88	65.69	76.98
S/16	1	58.90	70.01	73.10	**80.10**
S/14	2	**60.15**	**71.07**	**77.06**	77.21
S/14-Reg	2	59.63	69.88	74.16	77.99
B/8	1	**59.25**	70.79	**78.40**	**81.83**
B/16	1	58.53	**70.96**	67.23	81.07
B/14	2	55.25	69.67	75.84	77.62
B/14-Reg	2	54.90	69.36	77.67	76.29

The principle of "larger is better" applies here, where larger backbone sizes generally result in lower EERs. Utilizing all tokens yields significantly better results than relying solely on the CLS token. Moreover, utilizing multiple blocks performs better than using a single block, and the performance further improves with larger values of k. However, training downstream modules becomes more challenging as k increases, leading to convergence issues in some cases (denoted as "Failed"). Concatenating features yields better results than utilizing a weighted sum. Across similar sizes, there is generally no discernible difference in performance between DINO and DINOv2. Notably, for DINO, there is no clear performance distinction between using large and small patch sizes.

We selected the DINOv2 - ViT-L/14-Reg (chosen for its balance between performance and model size) to assess potential enhancements. Simple linear adaptors were utilized to reduce feature dimensionality and enable feature concatenation. Additionally, dropout was applied to mitigate overfitting. The results are presented in Table 5. The optimal configuration involves using dropout alongside linear adaptors and feature concatenation.

We applied the optimal configuration to EfficientNetV2, DeiT III, and EVA-CLIP and then compared their performances with that of DINOv2. The results are displayed in Table 6. DINOv2 clearly outperformed EfficientNetV2 and DeiT III, and it surpassed EVA-CLIP despite the latter's pre-training on a larger dataset with rich annotations (image-text pairs). These results underscore the advantage of using SSL for pre-training, enabling the learning of superior representations applicable to multiple tasks.

5.2. Approach 2: Fine-tuning final transformer blocks

We selected DINOv2 - ViT-L/14-Reg, extensively studied in **Approach 1**, as the representative of DINOv2. Similarly, we chose EfficientNetV2, DeiT III, and EVA-CLIP

Table 4. EERs of models with **Approach 1** utilizing various versions and architectures of DINO as backbones on the main (seen) test set. "Failed" indicates that the models failed to converge during training.

Model	Back-bone params	CLS token Final block	CLS token 4 blocks WS	CLS token 12 blocks WS	CLS token 4 blocks concat	CLS token 12 final blocks concat	All tokens Final block	All tokens 4 blocks WS	All tokens 8 blocks WS
DINO									
ViT-**S**/8	21M	26.43	21.16	21.23	20.41	18.72	16.03	Failed	Failed
ViT-**S**/16	21M	22.73	20.22	19.91	19.97	18.21	13.99	14.35	14.22
ViT-**B**/8	85M	**19.85**	**18.67**	**18.03**	**18.17**	**16.40**	14.09	13.87	17.74
ViT-**B**/16	85M	20.62	19.52	19.24	18.43	17.35	**13.95**	**13.52**	**13.95**
DINOv2									
ViT-**S**/14	21M	23.25	22.99	21.81	20.28	18.76	14.63	14.53	Failed
ViT-**S**/14-Reg	21M	26.78	23.16	23.05	20.61	18.92	15.02	15.26	Failed
ViT-**B**/14	86M	23.08	18.41	18.44	17.02	16.39	13.44	13.29	Failed
ViT-**B**/14-Reg	86M	23.69	20.27	20.08	19.63	19.32	14.37	13.84	Failed
ViT-**L**/14	300M	21.72	**16.84**	16.39	**15.82**	14.51	13.01	13.09	Failed
ViT-**L**/14-Reg	300M	22.43	18.97	16.88	18.75	15.19	14.00	12.67	12.64
ViT-**G**/14	1,100M	20.82	19.48	16.17	18.77	14.72	**11.67**	12.48	**11.72**
ViT-**G**/14-Reg	1,100M	**19.81**	18.02	**15.68**	17.76	**14.18**	12.40	**12.12**	12.46

Table 5. Enhancements to **Approach 1** utilizing SSL pre-trained DINOv2 - ViT-L/14-Reg as the backbone, with "L" denoting linear adaptors. Accuracies were calculated using a threshold of 0.5.

Blocks	Dropout	Fusion	Accuracy	EER	HTER
1	No	–	85.48	14.00	14.28
1	**Yes**	–	86.26	13.57	13.97
4	No	WS	84.15	12.67	15.35
4	**Yes**	WS	85.31	12.38	14.27
4	No	L+concat	86.52	13.41	13.35
4	**Yes**	L+concat	**87.42**	**11.98**	**12.41**

Table 6. Comparison between different ConvNet and transformer architectures on **Approach 1**. The final setting in Table 5 was applied. Accuracies were calculated using a threshold of 0.5.

Architecture	Accuracy	EER	HTER
EfficientNetV2 Large	78.67	21.77	21.45
DeiT III L/16-LayerScale	79.96	19.77	19.96
EVA-02-CLIP-L/14	83.31	16.51	16.76
DINOv2 ViT-L/14-Reg	**87.42**	**11.98**	**12.41**

for comparison. The performances of fine-tuning the final block (and also the tokens in the case of transformers) are shown in Table 7. Compared to **Approach 1**, all models gained better results, and the performance gaps between DINOv2 and others were reduced, with EVA-CLIP being the closest competitor. Nevertheless, DINOv2 remained the top performer. To narrow the gap with DINOv2, EVA-CLIP would need to be pre-trained with a vast dataset featuring rich annotations—a costly endeavor compared to DINOv2, which was pre-trained on a substantially smaller dataset without any annotations. Given the same architecture (DeiT III and DINOv2), the performance gap is nearly 6% in terms of EER. In addition to different training receipts, part of this drop may stem from the larger training data (although without annotations). Overall, these results again underscore the significant advantage of using SSL for pre-training ViTs.

Table 7. Comparison between different ConvNet and transformer architectures on **Approach 2**. The final block of each pre-trained backbone was fine-tuned for deepfake detection. Accuracies were calculated using a threshold of 0.5.

Architecture	Accuracy	EER	HTER
EfficientNetV2 Large	84.76	15.05	15.21
DeiT III L/16-LayerScale	82.64	17.21	17.28
EVA-02-CLIP-L/14	88.29	11.77	11.77
DINOv2 ViT-L/14-Reg	**88.78**	**11.32**	**11.26**

We next conducted an ablation study to determine the optimal number of k final blocks required for fine-tuning. It is important to note that different DINO backbone sizes have varying numbers of blocks; for example, DINOv2 - ViT-L/14-Reg has 24 blocks. The results are visualized in Fig. 2. If k is small, the model may not have enough parameters to adequately adapt to the new task, resulting in underfitting. Conversely, the model may experience overfitting, particularly when the fine-tuning dataset is small. The optimal number of k is approximately half of the total blocks, which in this experiment is 11. When increasing k from 1 to 11, the EERs decrease from 11.32% to 5.63%, representing an improvement of 5.69%.

5.3. Cross-dataset detection

In this experiment, we assessed the generalizability of the detectors in detecting unseen deepfakes. The scenario presented a robust challenge, as there were no diffusion images in the training set. The classification thresholds were recalibrated using the unseen validation set. The results are presented in Table 8. Notably, there were drops in the performance of all models, with the best one going from 11.32% to 27.61% in terms of EER. Overall, **Approach 2** consistently outperformed **Approach 1**. Within **Approach 2**, EfficientNetV2 exhibited better generalizabil-

Table 8. Comparison of performance between various ConvNet and transformer architectures on the unseen test set, comprising images generated or manipulated by diffusion-based methods. In **Approach 1**, "blocks" denotes the number of final blocks utilized for feature extraction, while in **Approach 2**, "blocks" signifies the number of fine-tuned blocks.

Model	Blocks	Threshold	Real	Repaint P2	Repaint LDM	LaMa	Pluralistic	Acc.	TPR	TNR	EER	HTER
Approach 1												
EfficientNetV2 Large	4	0.6355	47.89	52.89	55.78	49.11	56.67	52.47	47.89	53.61	49.22	49.25
DeiT III L/16-LayerScale	4	0.9983	52.89	52.67	56.44	44.44	52.67	51.82	52.89	51.56	47.83	47.78
EVA-02-CLIP-L/14	4	0.0737	63.44	43.00	52.44	31.22	51.22	48.27	63.44	44.47	45.75	46.04
DINOv2 ViT-L/14-Reg	4	0.0759	60.44	53.00	66.89	43.11	70.89	58.87	60.44	58.47	**40.67**	**40.54**
Approach 2												
EfficientNetV2 Large	1	0.5479	63.00	50.22	53.89	65.78	64.89	59.56	63.00	58.69	39.58	39.15
DeiT III L/16-LayerScale	1	0.9999	56.00	58.56	69.56	39.56	69.56	58.64	56.00	59.31	42.56	42.35
EVA-02-CLIP-L/14	1	0.9999	45.44	71.11	83.44	12.22	82.11	58.87	45.44	62.22	45.20	46.17
DINOv2 ViT-L/14-Reg	1	0.9980	50.78	70.22	78.22	65.00	86.78	70.20	50.78	75.06	**36.28**	**37.08**
DINOv2 ViT-L/14-Reg	11	0.7418	70.22	53.22	73.22	93.00	74.56	72.84	70.22	73.50	**27.61**	**28.14**

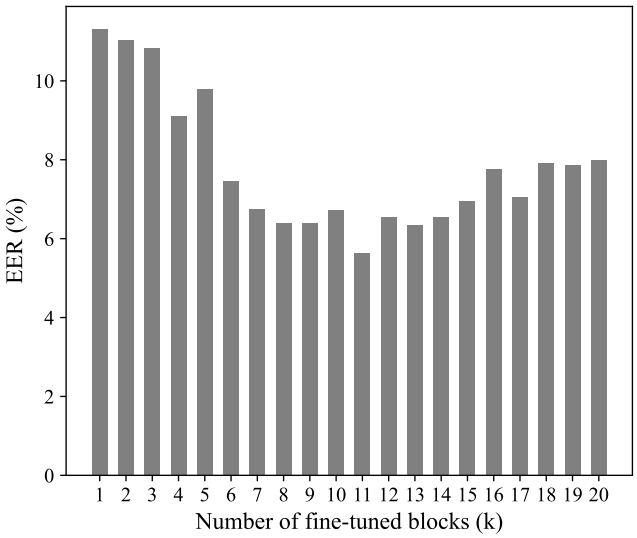

Figure 2. Results of ablation study for **Approach 2**. The numbers of fine-tuned final transformer blocks (k) and their corresponding EERs are shown. SSL pre-trained DINOv2 - ViT-L/14-Reg was used as the backbone.

ity compared to other supervised pre-trained transformers. DINOv2 maintained its position as the top performer, reaffirming the superiority of using SSL with ViTs.

5.4. Visualization and explainability

With **Approach 2**, we can naturally visualize the focus areas of the DINO-based model using attention weights. To simplify the process, we computed the average of the attention maps from all attention heads directed toward the CLS token. The results are depicted in Fig. 3. To underscore the efficacy of fine-tuning, we compared the outcomes with those of the corresponding frozen original model. The partially fine-tuned model primarily directed its attention to the forehead, eyes, nose, and mouth to assess the authenticity of the input image. This behavior closely mirrors human intuition in deepfake detection, as deepfake artifacts frequently manifest in these regions. Notably, the original version of DINO did not possess this ability. Even when presented with unseen deepfakes, the fine-tuned model consistently prioritized these areas. This explains the model's failure to detect deepfakes generated by Repaint-LDM, where the modification occurs in the hair region. In summary, such visualizations play a crucial role in deepfake detection, enhancing the interpretability of the results. The partially fine-tuned DINO model excelled in this regard.

6. Conclusion and future work

In this study, we explored two strategies for utilizing SSL pre-trained ViTs, specifically DINOs, as feature extractors for deepfake detection. The first approach involved utilizing frozen ViT backbones to extract multi-level features, while the second approach entailed partial fine-tuning on the final k blocks. Through extensive experimentation, we found that the fine-tuning approach demonstrated superior performance and interpretability, particularly through attention mechanisms to visualize the focused areas. Our findings provide valuable insights for the digital forensic community regarding the utilization of SSL pre-trained ViTs as feature extractors, a relatively underexplored area in the literature of deepfake detection.

Future work will primarily concentrate on forensic localization using DINOs without utilizing segmentation ground-truths during training. Additionally, efforts will be directed toward enhancing the generalizability of the models and exploring the potential of SSL on unlabeled deepfake datasets.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grants JP21H04907 and JP24H00732, and by JST CREST Grants JPMJCR18A6 and JPMJCR20D3, and by JST AIP Acceleration Grant JPMJCR24U3 Japan.

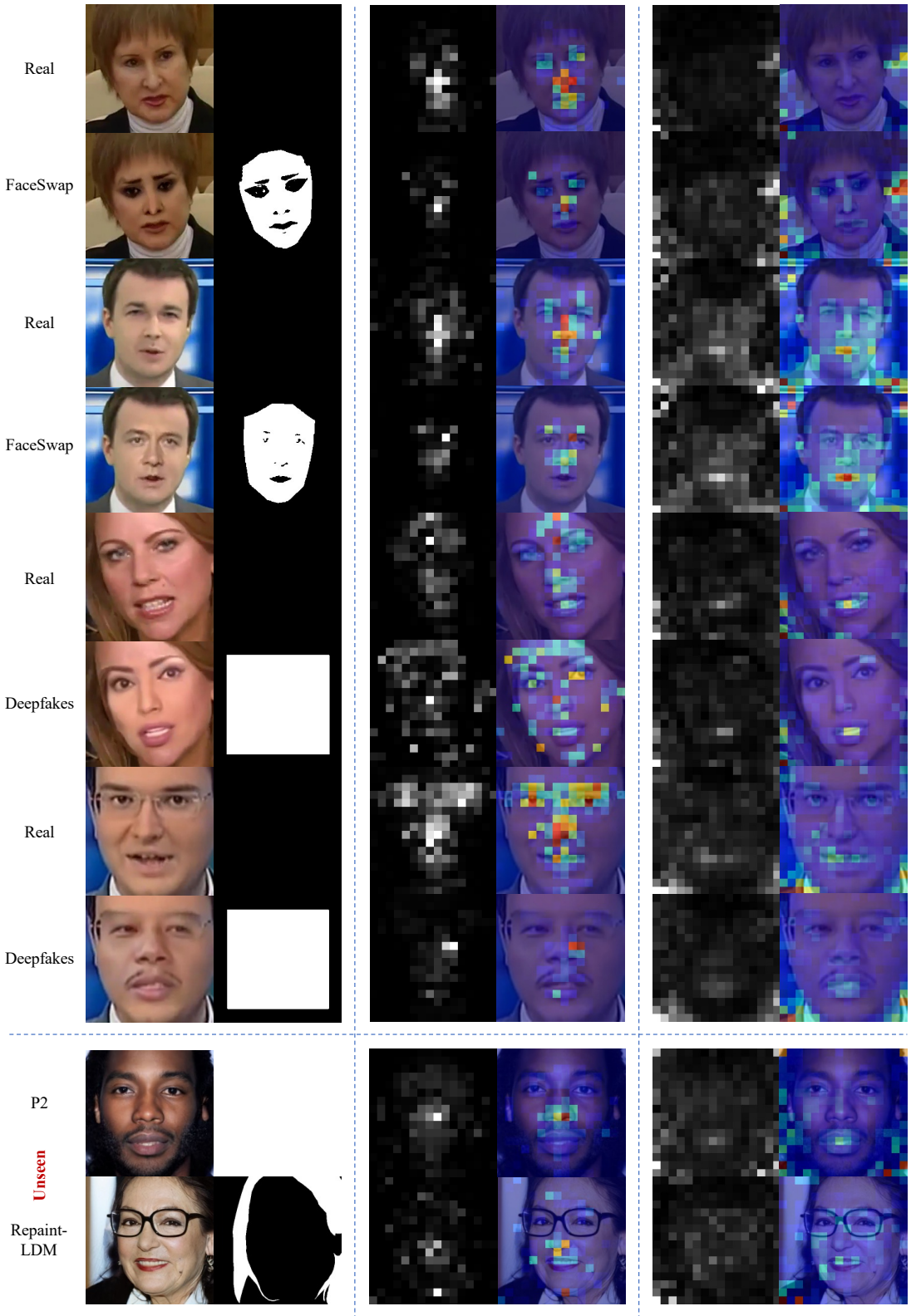

Figure 3. Visualization of averages of final block's multi-head attention maps of the partially fine-tuned DINOv2 - ViT-L/14-Reg from **Approach 2**, compared with those from its original pre-trained version. The training dataset includes Deepfakes and FaceSwap, while P2 and Repaint-LDM are unseen methods. Best viewed in color.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] M. Byeon, B. Park, H. Kim, S. Lee, W. Baek, and S. Kim. COYO-700M: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022. 4

[3] J. P. Cardenuto, J. Yang, R. Padilha, R. Wan, D. Moreira, H. Li, S. Wang, F. Andaló, S. Marcel, A. Rocha, et al. The age of synthetic realities: Challenges and opportunities. *APSIPA Transactions on Signal and Information Processing*, 12(1), 2023. 1

[4] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in NeurIPS*, 33:9912–9924, 2020. 2

[5] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 1, 2, 3, 4

[6] J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon. Perception prioritized training of diffusion models. In *CVPR*, pages 11472–11481, 2022. 4

[7] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018. 4

[8] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8188–8197, 2020. 4

[9] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1251–1258, 2017. 2

[10] J. S. Chung, A. Nagrani, and A. Zisserman. VoxCeleb2: Deep speaker recognition. In *INTERSPEECH*, pages 1086–1090, 2018. 4

[11] F. Cocchi, L. Baraldi, S. Poppi, M. Cornia, L. Baraldi, and R. Cucchiara. Unveiling the impact of image transformations on deepfake detection: An experimental analysis. In *ICIAP*, pages 345–356. Springer, 2023. 1, 2, 3, 5

[12] D. A. Coccomini, N. Messina, C. Gennaro, and F. Falchi. Combining efficientnet and vision transformers for video deepfake detection. In *ICIAP*, pages 219–229. Springer, 2022. 2

[13] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski. Vision transformers need registers. In *ICLR*, 2024. 1, 2, 4

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009. 4

[15] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer. The deepfake detection challenge (DFDC) dataset. *arXiv preprint arXiv:2006.07397*, 2020. 4

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 2

[17] N. Dufour and A. Gully. Contributing data to deepfake detection research. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html, 9 2019. 4

[18] J. Guan, H. Zhou, Z. Hong, E. Ding, J. Wang, C. Quan, and Y. Zhao. Delving into sequential patches for deepfake detection. *Advances in NeurIPS*, 35:4517–4530, 2022. 2

[19] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 2

[20] Y.-J. Heo, W.-H. Yeo, and B.-G. Kim. Deepfake detection algorithm based on improved vision transformer. *Applied Intelligence*, 53(7):7512–7527, 2023. 2

[21] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[22] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 4

[23] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 4

[24] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, pages 8110–8119, 2020. 4

[25] J. D. M.-W. C. Kenton and L. K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 1

[26] S. A. Khan and H. Dai. Video transformer for deepfake detection with incremental learning. In *ACM MM*, pages 1821–1828, 2021. 2

[27] P. Korshunov and S. Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018. 4

[28] I. Kukanov, J. Karttunen, H. Sillanpää, and V. Hautamäki. Cost sensitive optimization of deepfake detector. In *APSIPA ASC*, pages 1300–1303. IEEE, 2020. 4

[29] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *CVPR*, pages 3207–3216, 2020. 4

[30] H. Lin, W. Huang, W. Luo, and W. Lu. Deepfake detection with multi-scale convolution and vision transformer. *Digital Signal Processing*, 134:103895, 2023. 2

[31] H. Liu, Z. Tan, C. Tan, Y. Wei, Y. Zhao, and J. Wang. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *CVPR*, 2023. 2, 3

[32] H. H. Nguyen, J. Yamagishi, and I. Echizen. How close are other computer vision tasks to deepfake detection? In *IJCB*, pages 1–10. IEEE, 2023. 4, 5

[33] U. Ojha, Y. Li, and Y. J. Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, pages 24480–24489, 2023. 2, 3

[34] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023. 1, 2, 4

[35] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 1, 2, 4

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

[37] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung. Deepfake detection: A systematic literature review. *IEEE access*, 10:25494–25513, 2022. 1, 3

[38] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 4

[39] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, pages 1–11, 2019. 4

[40] C. Sanderson and B. C. Lovell. Multi-region probabilistic histograms for robust and scalable identity inference. In *ICB*, pages 199–208. Springer, 2009. 4

[41] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. *Advances in NeurIPS*, 35:25278–25294, 2022. 4

[42] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 4

[43] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, pages 2149–2159, 2022. 4

[44] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR, 2019. 2

[45] M. Tan and Q. Le. Efficientnetv2: Smaller models and faster training. In *ICML*, pages 10096–10106. PMLR, 2021. 4

[46] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64:131–148, 2020. 1, 3

[47] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 2

[48] H. Touvron, M. Cord, and H. Jégou. DeiT III: Revenge of the VIT. In *European conference on computer vision*, pages 516–533. Springer, 2022. 4

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in NIPS*, 30, 2017. 1

[50] J. Wang, Z. Wu, W. Ouyang, X. Han, J. Chen, Y.-G. Jiang, and S.-N. Li. M2tr: Multi-modal multi-scale transformers for deepfake detection. In *ICMR*, pages 615–623, 2022. 2

[51] T. Wang, H. Cheng, K. P. Chow, and L. Nie. Deep convolutional pooling transformer for deepfake detection. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(6):1–20, 2023. 2

[52] P.-W. Wu, Y.-J. Lin, C.-H. Chang, E. Y. Chang, and S.-W. Liao. RelGAN: Multi-domain image-to-image translation via relative attributes. In *ICCV*, pages 5914–5922, 2019. 4

[53] C. Zhao, C. Wang, G. Hu, H. Chen, C. Liu, and J. Tang. Istvt: interpretable spatial-temporal video transformer for deepfake detection. *IEEE Transactions on Information Forensics and Security*, 18:1335–1348, 2023. 2

[54] C. Zheng, T.-J. Cham, and J. Cai. Pluralistic image completion. In *CVPR*, pages 1438–1447, 2019. 4

[55] D.-C. Țânțaru, E. Oneață, and D. Oneață. Weakly-supervised deepfake localization in diffusion-generated images. In *WACV*, pages 6258–6268, 2024. 4