FoundPAD: Foundation Models Reloaded for Face Presentation Attack Detection

Guray Ozgur[1], Eduarda Caldeira[1], Tahar Chettaoui[1], Fadi Boutros[1], Raghavendra Ramachandra[3], Naser Damer[1,2]

[1] Fraunhofer IGD, Germany, [2] TU Darmstadt, Germany
[3] Norwegian University of Science and Technology (NTNU), Norway
Email: guray.ozgur@igd.fraunhofer.de

Abstract

Although face recognition systems have seen a massive performance enhancement in recent years, they are still targeted by threats such as presentation attacks, leading to the need for generalizable presentation attack detection (PAD) algorithms. Current PAD solutions suffer from two main problems: low generalization to unknown scenarios and large training data requirements. Foundation models (FM) are pre-trained on extensive datasets, achieving remarkable results when generalizing to unseen domains and allowing for efficient task-specific adaption even in low data availability settings. This is one of the first works to recognize the potential of FMs and adapt them for the downstream task of PAD. The FM under consideration is adapted with LoRA weights while simultaneously training a classification header. The resultant architecture, FoundPAD, is highly generalizable to unseen domains, achieving competitive results in several settings under different data availability scenarios and even when using synthetic training data. To encourage reproducibility and facilitate further research in PAD, we publicly release the implementation of FoundPAD at https://github.com/gurayozgur/FoundPAD.

1. Introduction

Face recognition (FR) systems have seen a massive performance improvement in recent years, mainly due to the development of deep learning [5, 13]. However, they still suffer from different malicious attacks [37]. Presentation attacks (PA) constitute an example of such threats, as they allow the attacker to claim an identity different from their own through techniques such as 3D masks, printed images and replayed videos [4, 17, 65, 66], among others. When left undetected, these attacks can lead to several dangerous situations, such as identity theft [37] or unauthorized access to confidential information, as they make the attacker able to impersonate another identity. To tackle this issue several presentation attack detection (PAD) systems have been proposed [15, 16, 27, 34, 45, 46, 55, 58, 60, 61]. These systems classify face images as unaltered samples (bona-fide samples) or PAs, allowing the detection of attacks before they are used to verify identities in critical processes. Although these systems can achieve high performance in the intra-dataset scenario [16, 34, 60, 61], the high sample variability in the cross-domain scenario results in a significant domain change, leading to the need for developing techniques that specifically address this problem [15, 27, 45–47, 55, 58]. Another challenge faced by PAD systems is the need for large training datasets to achieve good performance levels [18, 65]. The performance of PAD networks in low data availability scenarios where smaller training datasets are usually limited to a reduced number of data distributions generally results in less generalizable models [15].

Foundation models (FM) contain many trainable parameters pre-trained with self-supervised learning. This learning paradigm allows FMs to learn from unlabeled data, which lifts some common constraints usually associated with model training, namely the difficulty in creating large-scale training datasets. Hence, FMs can be trained on extensive and diverse datasets, achieving high generalizability for a wide range of tasks [3]. Although initially FMs were mainly deployed in natural language processing (NLP) tasks, they have also been effectively used to address computer vision tasks [30, 39, 42, 43]. In biometrics, FM utilization is still in an early exploratory stage, with very recent works addressing synthetic face image generation [41], iris segmentation [20] and FR [11]. Biometrics tasks such as cross-domain PAD are expected to highly benefit from the generalization power of FMs, namely in challenging scenarios with low data availability, where they have outperformed models trained from scratch [11]. However, this is still an unexplored path for addressing the PAD task.

This work takes advantage of FMs potential to perform generalizable PAD with low data requirements. In particular, we adapt the pre-trained FM CLIP [42] to the PAD task using a low-rank adaption (LoRA), which allows the network to adapt its feature space to the downstream PAD task without losing the knowledge acquired during its self-

supervised pre-training, while simultaneously training a header to perform classification. This adapted FM corresponds to our proposed framework, FoundPAD. To show that FoundPAD is effectively taking advantage of FMs' properties, we further propose and evaluate three alternative FM and transformer-based methods. First, we assess the importance of the FM's pre-trained weights by comparing FoundPAD with models following the same architectures trained from scratch (ViT-FS). Then, we prove the importance of properly adapting the FM to the considered downstream task by assessing CLIP's zero-shot performance on PAD (TI) and evaluating a training scenario where no LoRA adaptation is performed and only the classification layer is trained (FE). With these experiments, we were able to prove FoundPAD's superiority in taking advantage of FMs' potential to perform PAD, with FoundPAD achieving an average single-source HTER lower than the second best-performing method in the literature by 6.54 percentage points (pp.) while surpassing ViT-FS and FE by 4.35 pp. and 8.94 pp., respectively, for CLIP ViT-L. Hence, this work contributes to a better understanding of the generalization capabilities of FMs, through our FoundPAD, proving their capability to surpass previously existent SOTA PAD solutions in many of the considered benchmarking protocols.

2. Related Work

Cross-Domain PAD: The recent deep learning advances have boosted the development of high-performing PAD solutions [16,34,47,60,61], especially in intra-dataset evaluation scenarios. However, PAs can take several forms e.g., printed images, replayed videos, and 3D masks, and the samples used for model training can be acquired under different conditions for distinct datasets, resulting in a significant domain shift when assessing the performance of PADs under the cross-dataset scenario. This results in degraded performance of the SOTA intra-dataset PAD models, leading to the need for developing techniques that perform well across different domains [15,27,45,46,55,58]. These methods can be grouped into domain adaption (DA) and domain generalization (DG) strategies. DA methods [31, 52] train PAD networks with both labelled source and unlabelled target domain data to learn a discriminative feature space that can be used to perform PAD across different domains efficiently. Since only the labels of the source domain are considered, knowledge is transferred to the target domain by aligning its feature space with the one produced by the source features, using techniques like maximum mean discrepancy minimization [31] and adversarial training [52]. Nonetheless, two main problems arise when considering DA-based PAD strategies. First, DA-based PAD requires collecting target domain data, which is often difficult and time-consuming. Second, even if this data is available, using it during the training process is not representative of real-world scenarios, where the target domains may not be completely known, making access to their data usually inexistent during training [15]. To avoid relying on target testing data, DG strategies [10, 15, 27, 45, 46] use training data from several sources simultaneously to enable a broader understanding of distinct domains by jointly analysing their data distributions. DG-based PAD methods follow different approaches, namely adversarial training [27, 45] and meta-learning [10, 46]. Although these approaches have achieved good results in unseen target domains, they rely on the availability of labelled data from several sources, which is challenging to satisfy in practice, and sometimes rely upon multi-stage or multi-network training strategies [10, 33], which induce high computational costs.

Low Data Availability: In addition to cross-domain issues, the necessity of large PAD training datasets has been brought up in several works [18, 65]. Even though the main two available large-scale datasets, CelebA-Spoof [65] and SynthASpoof [18], have crossed the threshold of 2k bonafide and 4k attack samples, there is still a lack of large-scale PAD datasets. This is particularly problematic in cross-domain PAD, where smaller training datasets are usually limited to a small number of domains, resulting in less generalizable models [15].

Foundation Models: FMs are built with a vast number of trainable parameters, enabling their training on extensive and diverse datasets, which is particularly beneficial for fields that tackle a wide variety of tasks, such as computer vision [11, 20, 30, 39, 41, 42]. Kirillov *et al.* [30] designed the Segment Anything Model (SAM) for image segmentation across various domains, achieving a remarkable generalization capacity that makes it able to handle novel image distributions. DINOv2 networks [39] are self-supervised pre-trained visual models capable of generating universal features for image-level and pixel-level tasks. Radford *et al.* [42] introduced Contrastive Language-Image Pretraining (CLIP), a multimodal FM trained to process visual and textual inputs, thus learning the relationships between them.

Although the extensive training data used by vision FMs grants them a high degree of generalizability, their performance often falls short in specialized settings [50]. As a result, several techniques that adapt Vision Transformer (ViT) networks to specific downstream tasks have been proposed [8, 9, 22]. ViT-Adapter [9] achieved SOTA performance on the COCO dataset by incorporating fine-grained multi-scale feature reconstruction and embedding image-specific inductive biases into the FM. AdaptFormer [8] replaced the standard MLP block in the transformer encoder with two parallel MLP branches, one that mirrors the original network, preserving its generalization capabilities, and another for task-specific fine-tuning. Hu *et al.* [22] defined the concept of LoRA layers, which consist of trainable rank decomposition matrices that are inserted in the pre-trained FM. When new data is fed to this network, its weights are

frozen and only the LoRA weights are fine-tuned, allowing to adapt the FM to a new task without disregarding its previously acquired knowledge. LoRA has proven effective across diverse applications, namely capsule endoscopy diagnosis [62], plant phenotyping [7], and FR [11]. In this study, we select LoRA to adapt CLIP for the PAD task, given its promising results in biometrics [11].

Despite the increasing attention FMs have received in recent years, their application in biometrics remains largely underexplored. [41] used FMs to synthesize facial images conditioned on identity-specific information. [20] adapted SAM [30] to perform iris segmentation. A recent work [11] adapted DINOv2 [39] and CLIP [42] to the FR task with LoRA [22]. The results showcase the benefits of adapting FMs with LoRA, as the adapted networks showed competitive performances to models trained from scratch, outperforming them in challenging scenarios with low data availability. As will be proved later in Section 5, FMs can also boost PAD performances in low data availability settings.

In this work, we recognize that FMs can be particularly helpful in addressing tasks usually associated with generalization problems, such as cross-domain PAD. Simultaneously, FMs may be greatly beneficial when applied to the challenging low data availability scenario that commonly affects PAD [11].

3. Methodology

3.1. CLIP: A preliminary

CLIP [42] is a multimodal FM designed to process simultaneously visual and textual inputs. In this work, we select it as the base FM to perform PAD due to its competitive results in zero-shot learning scenarios [42] and generalization capacity across a wide range of tasks [42], including specific downstream tasks when adapted with LoRA [11], which is of high importance in domain-specific settings as those imposed by PAD. CLIP was trained on a vast dataset of paired image and text samples, enabling it to learn the connections between these two modalities. Its architecture employs two separate encoders (one for images and another for text) trained simultaneously using a contrastive learning objective that measures the cosine similarity between their output features. For positive pairs, where the text matches the image, the similarity is maximized, while for negative pairs, where they do not correspond, it is minimized. This training strategy enables CLIP to effectively capture the semantic relationships between images and their descriptions, resulting in a highly versatile model capable of generalizing across diverse tasks [42]. Furthermore, CLIP demonstrates strong performance in zero-shot learning scenarios, where no task-specific fine-tuning is required.

In this work, we leverage CLIP to process image-text pairs and single-image inputs in various approaches to rationalise our FoundPAD. Initially, we assess CLIP's potential as a PAD solution by evaluating its ability to distinguish between bona-fide and PA samples without additional training, as described in Section 3.3. Towards that, CLIP processes image-text pairs where the text describes the possible labels of the image ("biometric presentation attack" and "bona-fide presentation" following ISO/IEC 2382-37 [26]). In other approaches, we use only the image encoder, treating CLIP as a feature extractor. A binary classification layer is added on top of the extracted features to perform PAD, following recent works that successfully applied FMs to downstream tasks such as image segmentation [30] and FR [11]. In these approaches, later detailed in Sections 3.2 and 3.3, we initialize CLIP with the pre-trained weights made publicly available in [42][1].

3.2. FoundPAD

In this work, we propose to adapt FMs to the downstream task of PAD, taking advantage of their high capacity to generalize to novel domains. To this end, we propose a framework that adapts the pre-trained FM with LoRA layers, FoundPAD, shifting the generated feature space in a direction that facilitates samples' classification as bona-fide or PAs. As discussed in Section 3.1, some FMs, such as CLIP, consist of two main components: a text encoder and an image encoder. This dual architecture allows CLIP to classify images into a specific group of categories, by defining a text input that describes each of them. In the PAD scenario, these textual inputs simultaneously describe the two possible labels, "biometric presentation attack" and "bona-fide presentation" [26]. However, FMs can also process images without any textual input, functioning as a feature extraction tool that can be combined with a classification layer for task-specific classification. In this study, we investigate the FMs' effectiveness as a feature extractor for PAD by removing the text encoder and adapting the image encoder to extract relevant features to perform PAD.

Fine-Tuninig with LoRA: The image encoder of FMs such as CLIP, is composed of alternating multi-headed self-attention (MSA) layers and multi-layer perceptron (MLP) blocks, with layer normalization applied before each block and residual connections following each block. While FMs can be used for PAD out-of-the-box without fine-tuning, this may lead to suboptimal performances since the produced embedding space is not necessarily optimal for PAD, as will be shown in our detailed experiments. However, training large FMs from scratch would result in losing their FM properties, as we will also show in Section 5. Hence, we opt to incorporate a ViT adapter [8, 9, 22] to fine-tune CLIP, as described in Section 2. In particular, we use LoRA [22] for this purpose, due to its ability to adapt FMs to highly specific-domain tasks [7, 62] and given its strong performance in related fields, such as FR [11].

LoRA leverages low-dimensional reparametrization, which has been proven to be as effective as training the full

[1] https://github.com/OpenAI/CLIP

parameter space [1] while significantly reducing the number of trainable parameters. When using this technique, CLIP's original weights are frozen, and a set of trainable rank-decomposition matrices is introduced into each layer of the transformer architecture, enabling efficient adaption with minimal parameter updates. Given a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, the low-rank decomposition introduced by LoRA updates it as:

$$W_0 + \Delta W = W_0 + \gamma_r BA \tag{1}$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ are the trainable rank-decomposition matrices, with the rank $r << min(d, k)$, and γ_r is a scaling factor. While γ_r was originally defined as $\frac{\alpha}{r}$, the constant α tends to cause gradient collapse as r increases, leading to an absence of performance improvements for higher ranks [28], although more trainable parameters are used for fine-tuning. This problem can be fixed by using rank-stabilized LoRA (rsLoRA) [28], which scales BA with $\frac{\alpha}{\sqrt{r}}$ instead of $\frac{\alpha}{r}$, allowing higher ranks to perform better due to the absence of gradient collapse. Hence, we opt to use rsLoRA to fine-tune CLIP, by setting $\gamma_r = \frac{\alpha}{\sqrt{r}}$. As previously mentioned, W_0 is kept frozen, and, since γ_r is constant, only A and B are updated during fine-tuning. After the fine-tuning process is complete, the final model weights, W, are computed by adding the original weights to the LoRA weights, $W = W_0 + \gamma_r BA$, which does not introduce additional parameters during inference, preserving the computational efficiency of the original model while benefiting from the fine-tuning improvements.

To ensure efficiency and reduce the number of parameters in the proposed approach, LoRA is applied only to the MSA weights, leaving the MLP blocks unchanged [22]. While LoRA can be applied to the projection matrices of the query, key, value, and output (q, k, v, and o, respectively) in the MSA, we limit the adaption to the q and v matrices. This choice follows the recommendations from the original LoRA paper [22] and a recent application based on FMs in the biometrics field [11]. The MSA mechanism involves h parallel attention heads, each with its own set of q, k, and v matrices. Each head is adapted independently with LoRA, resulting in distinct LoRA weights for each of them. When an embedding x is processed through the MSA, the projections Q_i, K_i, and V_i (for the query, key, and vector, respectively) in head i are determined as follows:

$$\begin{aligned} Q_i &= W_i^q x + \gamma_r B_i^q A_i^q x \\ K_i &= W_i^k x \\ V_i &= W_i^v x + \gamma_r B_i^v A_i^v x \end{aligned} \tag{2}$$

where W_i^q, W_i^k and W_i^v are the frozen projection layers for q, k and v, respectively, and A_i^q, B_i^q, A_i^v and B_i^v correspond to the trainable LoRA layers' parameters. The attention score for head i, denoted as $Attention(Q_i, K_i, V_i)$, can then be computed as shown in Equation 3:

$$Attention(Q_i, K_i, V_i) = Softmax\Big(\frac{Q_i K_i^T}{\sqrt{d_k}}\Big)V_i \tag{3}$$

where the scaling factor d_k represents the dimension of the key vectors. The MSA layer's output is generated by the projection layer O, which takes the concatenated attention scores from all heads along the feature axis as input:

$$Multihead(Q, V, K) = Concat(head_1, ..., head_k)W^0 \tag{4}$$

The output from the MSA is then passed through the static MLP, completing the processing within a single ViT block. The output from ViT block l is subsequently fed into block $l + 1$, which consists of a new MSA fine-tuned with LoRA, followed by another fixed MLP.

Classification: Using LoRA to fine-tune the FMs results in a feature space adapted to the PAD task and, thus, is expected to yield better results. The classification is performed by an extra fully connected layer responsible for processing the features extracted by the adapted FM and output the final predictions, $\tilde{y}$. Finally, the binary cross-entropy loss is used to compare $\tilde{y}$ with the ground truth labels, y, allowing to update the model's trainable parameters:

$$L_{BCE} = -(y\, log(\tilde{y}) + (1 - y)\, log(1 - \tilde{y})) \tag{5}$$

During testing, FoundPAD's weights are frozen and the highest output score produced by the classification layer's neurons defines the model prediction for each sample.

3.3. Baselines

To analyse the usage of FMs to perform PAD in detail and prove the effectiveness of FoundPAD when compared with alternative baseline solutions, we present three alternative FM or transformer-based methods for PAD.

Text-Image (TI): FMs, such as CLIP, have demonstrated remarkable performance in zero-shot learning scenarios across various downstream tasks, including food classification, car model classification, and offensive memes identification [42]. These tasks involve the simultaneous use of text and image encoders for classification. To explore the zero-shot learning capabilities of the selected FM in the PAD task, we evaluated its performance using image-text pairs where the text describes the possible labels of the image. Specifically, each test sample was paired with textual descriptions of its possible labels: "biometric presentation attack" and "bona-fide presentation". The similarity score between the image embedding and the corresponding text embeddings determined the predicted label. This approach utilizes the complete FM architecture without requiring further training. However, due to the lack of task-specific adaptation and the domain-specific nature of PAD, the TI approach is anticipated to perform suboptimally compared to alternative scenarios or more general visual tasks (e.g. detecting if an image contains a cow).

ViT Trained from Scratch (ViT-FS): Part of the FMs' abilities is related to the architectures they are based

on (commonly ViT architectures [2]), which have shown promising results when trained for both PAD [23] and morphing attack detection [63]. Hence, we investigate if ViT networks can individually contribute to building a strong PAD, by training from scratch identical architectures to the ones used in the considered FMs (ViT-B and ViT-L), using only the selected PAD training dataset. Since the transformer parameters are randomly initialized, ViT-FS cannot be considered an FM, as it does not benefit from any knowledge acquired during any previous training. This allows for direct comparison between FM-based techniques such as FoundPAD and visual transformers, making it possible to assess how valuable the in-built knowledge of FMs is for downstream tasks such as PAD.

Feature Extractor (FE): To evaluate the impact of the network adaptation enabled by LoRA in FoundPAD, we design an experiment in which the FM is frozen without adaption and used as a feature extractor. PAD is then performed by a binary classification layer trained on top of the FM's feature space following Equation 5. This experiment allows us to evaluate if the FM's original feature space captures relevant information to perform PAD while providing a reference for quantitatively measuring the improvements resulting from adapting the model's weights with LoRA.

4. Experimental Setup

Datasets: To allow for a wide range of fair comparisons, experiments were conducted on five publicly available datasets widely used in cross-dataset PAD works to benchmark their performances [15, 16, 32, 33, 40, 45, 46]: MSU-MFSD [56] (denoted as M), CASIA-FASD [66] (denoted as C), Idiap Replay-Attack [12] (denoted as I), OULU-NPU [4] (denoted as O), and CelebA-Spoof [65] (denoted as CA). Given the adaption of synthetic data as a privacy-friendly alternative for authentic data in biometric development [18], we additionally use the synthetic-based face PAD dataset SynthASpoof [18] as a training dataset paired with the mentioned four datasets as evaluation benchmarks, following the protocol defined in [18]. Apart from bona-fide samples, the **MSU-MFSD** [56] dataset includes printed photo and replay attacks, totalling 440 videos from 35 subjects. The **CASIA-FASD** [66] includes 600 videos from 50 subjects and contains warped photo, cut photo and video replay attacks. The **Idiap Replay-Attack** [12] dataset consists of 300 videos from 50 subjects and includes both print and replay attacks. The **OULU-NPU** [4] is a mobile face PAD dataset and contains 5940 videos from 55 subjects, acquired with six distinct mobile phones. The **CelebA-Spoof** [65] is diverse in terms of subjects, illumination, and sensors and comprises four types of attacks, namely print, replay, 3D mask and paper cut attacks. Its images were collected from the web, resulting in a large-scale dataset with 625,537 images from 10,177 subjects. The **SynthASpoof** [18] dataset is designed to address privacy and scalability challenges in PAD research. It contains 25,000 subjects each having only one sample generated using StyleGAN2-ADA [29], filtered through CR-FIQA [6] to ensure high quality and realistic appearance. For print attacks, 3,800 subjects were printed and recaptured. Replay attacks were recorded by displaying synthetic images on two different screens and recapturing using three devices, where each device contributed 25,000 images adding up to a total of 75,000 attack images.

All the performed experiments target the cross-dataset scenario, meaning that samples from different datasets are used during training and testing. The number of datasets used for training should be taken into account, as models trained on data from different datasets can learn from distinct information sources and, thus, are expected to perform better in cross-dataset scenarios. Hence, the developed experiments are divided into three groups based on the scale of the data available for training and following established evaluation protocols: triple-source (3 training datasets), double-source (2 training datasets) and single-source (1 training dataset). We perform five triple-source experiments (training dataset(s) $\rightarrow$ testing dataset): O&C&I $\rightarrow$ M, O&M&I $\rightarrow$ C, O&C&M $\rightarrow$ I, I&C&M $\rightarrow$ O, O&C&M $\rightarrow$ CA, following previous works [10, 15, 27, 33, 53]. For the double-source scenario, two cases are considered: M&I $\rightarrow$ C and M&I $\rightarrow$ O [15, 16, 38, 60, 61]. The single-source scenario includes a set of twelve experiments where one of the M, C, I, and O datasets is used to train the network and the remaining three are separately used for testing, following previous works on cross-dataset PAD [15, 27, 46, 53, 54]. The SynthASpoof dataset is used to adapt models from the synthetic domain to the authentic domain. Following Fang *et al.* [18], models are trained only on the SynthASpoof dataset and then evaluated on M, C, I, and O.

Image Pre-Processing: Before being fed into the image encoder, the training images undergo a preprocessing procedure. We detect the face in each sample using MTCNN [64] and resize it to 256×256 pixels, following [15]. During training, all samples are also subject to the data augmentation process later defined in this section. Then, the images are processed in a way that allows them to be tokenized, given the success achieved with tokenization in FMs' NLP applications [2]. The tokenization process follows the procedure proposed by Alexey *et al.* [2]. First, the image is divided into non-overlapping regions, which are then processed by a linear projection layer to create patch embeddings. These patch embeddings are combined with a learnable class (CLS) token [2], forming a unified representation of the image that aids in classification. Position embeddings are also added to preserve the spatial order of the original image patches. The resulting embedding vector, enriched with patch-level information, position data, and the CLS token, is then used as the input for CLIP's image encoder [2].

Model Architecture: The used FM, CLIP [42], intro-

duced four models based on two distinct architectures: base and large. The base architecture, comprising 86M parameters, is available in two versions differing by patch size (16 and 32). In contrast, the large architecture has 0.3 billion parameters and includes a variant fine-tuned at a higher resolution of 336 pixels for one extra epoch to enhance performance [48]. Inspired by recent studies leveraging CLIP for FR [11], we select one model from each architecture for evaluation: the base model with a patch size of 16 and the large model trained without higher-resolution inputs, from now on referred to as ViT-B and ViT-L, respectively. These architectures were used in all our experiments, whether under the TI, ViT-FS, FE, or FoundPAD settings.

Implementation Details: All of the models presented in this study were trained for 40 epochs using the AdamW [35] optimizer with a momentum of 0.9 and a weight decay of 0.05, following [11]. The batch size was set to 512 for all experiments [11], except for ViT-L-FS, where a batch size of 432 was used due to GPU constraints. For FoundPAD, the LoRA r, α and dropout were set to 8, 8 and 0.4, respectively. For ViT-FS and FoundPAD, the learning rate of the ViT network was defined as 1e-6. The learning rate of the binary classification layer was set to 1e-3 for all experiments requiring training (ViT-FS, FE, and FoundPAD). The data was augmented using random crop to 224×224 pixels, random horizontal flip, random gamma correction with gamma limits 80 and 180, an RGB shift with a limit of 20 for each colour component and colour jitter (with brightness, contrast, saturation and hue set to 0.1), following [15].

Evaluation Metrics: Following previous work on cross-domain PAD [15, 16, 32, 33, 45, 46], the Half Total Error Rate (HTER) and the Area under the Receiver Operating Characteristic (ROC) Curve (AUC) value were determined in percentage (%) for the performed experiments. The HTER is defined as the mean of the standard PAD evaluation metrics Bona-fide Presentation Classification Error Rate (BPCER) [25] and Attack Presentation Classification Error Rate (APCER) [25].

5. Results and Discussion

Zero-Shot PAD (TI): As described in Section 3.3, each test sample was paired with textual descriptions of "biometric presentation attack" and "bona-fide presentation", with the similarity score between the image embedding and the text embeddings determining the predicted label. This evaluation's results (Table 1) reveal CLIP's limitations for PAD without training, with high HTERs and low AUCs, performing near random. ViT-L shows a slight improvement over ViT-B, but the overall performance remains poor. This is likely due to CLIP's text encoder, which lacks domain-specific semantics, struggles with nuanced PAD characteristics, and fails to align effectively with dataset features. These results highlight the need for domain-adapted fine-tuning or enhanced prompts for effective PAD using CLIP.

Baselines toward FoundPAD (ViT-FS and FE): Since using image-text pairs did not achieve any tangible results, we further explored the embedding space of the FM's image encoder, which is induced by the pre-trained weights of the model. To measure this quantitatively, ViT-FS and FE (Section 3.3) were evaluated under triple-source, double-source, and single-source scenarios with two different architectures (Tables 2, 4, and 5). With a higher data availability, i.e. the triple-source case, ViT-FS achieved better results than FE for ViT-B, while performing worse for ViT-L. Given the reduced size of the datasets used to train ViT-FS and FE, ViT-FS is likely to underperform when a larger architecture is considered, as it does not benefit from the FM's previous knowledge. Simultaneously, the number of trainable parameters in ViT-B is significantly smaller, which might justify ViT-FS's increased performance in this scenario. For ViT-B, ViT-FS had an average HTER of 10.37% and an AUC of 95.99%, while FE achieved an average HTER of 28.14% and an AUC of 80.32% (Table 2). In lower data availability settings, i.e. the double and single-source cases, the performance gap between ViT-FS and FE remained large. The double-source case in Table 4 shows that ViT-FS achieved an HTER of 14.00% and 7.11% on CASIA-FASD and OULU-NPU, respectively, while FE achieved 27.22% and 33.57% for these metrics when considering ViT-B. The same phenomenon is seen for the single-source case in Table 5 as HTER averages are 15.88% for ViT-FS and 31.89% for FE with the ViT-B. Similar gaps are also observed for ViT-L for the single-source case. Hence, while using the embedding space of the FM is a good start and can achieve better results than a binary CNN [59], it is by no means an optimal embedding space. This suggests that either the embedding space of the FM should be aligned to the PAD task or a deeper classification network should be used.

FoundPAD: Since the FMs' embeddings should be aligned to the PAD, we adapt the embedding space for PAD-specific nuances, leading to the FoundPAD framework described in Section 3. FoundPAD was evaluated on the same three scenarios as ViT-FS and FE. While comparing with ViT-FS allows for withdrawing conclusions regarding the power of FMs' pre-training on large-scale databases and further adaption to the available data, the comparison with FE reveals the re-usability of the original FM's embedding space and the benefits of adapting the FM with LoRA to the PAD downstream task. For the triple-source scenario (Table 2), FoundPAD achieved average HTER and AUC values of 10.62% and 95.52%, respectively, for ViT-B, and of 9.67% and 96.60%, respectively, for ViT-L. When considering the ViT-L architecture, these results reveal an improvement in HTER of 13.37 pp. and 9.09 pp. compared to ViT-FS and FE, respectively. For ViT-B, FoundPAD performed similarly to ViT-FS while surpassing FE by 17.52 pp. in terms of average HTER. For the double-source sce-

Table 1. Zero-shot learning (TI) results of the considered FM architectures, CLIP ViT-B and ViT-L, on five evaluation benchmarks. Both architectures present very limited classification capabilities, achieving close to random performance for most of the analysed scenarios.

Model	M		C		I		O		CA	
	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑
ViT-B	55.71	41.22	50.67	49.53	50.50	50.74	52.05	47.87	56.07	42.02
ViT-L	41.19	62.96	43.44	56.56	46.50	54.49	44.76	59.44	58.07	39.39

Table 2. Results of triple-source cross-dataset evaluation on four benchmarking datasets. The best and second-best results for each metric are highlighted in bold and underlined, respectively. It can be seen that FoundPAD surpasses both ViT-FS and FE in most of the evaluation scenarios while achieving comparable or even superior performances than SOTA PAD methods.

Method		O&C&I → M		O&M&I → C		O&C&M → I		I&C&M → O		Average	
		HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑
Binary CNN [57]		29.25	82.87	34.88	71.94	34.47	65.88	29.61	77.54	32.05	74.56
Auxiliary [34]		22.72	85.88	33.52	73.15	29.14	71.69	30.17	77.61	28.89	77.08
ResNet50-PS [60]		14.32	94.51	18.23	89.75	18.86	89.63	21.44	87.56	18.21	90.36
NAS-FAS [61]		19.53	88.63	16.54	90.18	14.51	93.84	13.8	93.43	16.10	91.52
LMFD [16]		10.48	94.55	12.50	94.17	18.49	84.72	12.41	94.95	13.47	92.10
ViTransPAD [38]		8.39	-	21.27	-	16.83	-	15.63	-	15.53	-
PatchNet [51]		7.10	**98.64**	11.33	94.58	14.60	92.51	11.82	95.07	11.21	95.20
MADDG [45]		17.69	88.06	24.50	84.51	22.19	84.99	27.89	80.02	23.07	84.40
RFM [46]		17.30	90.48	13.89	93.98	20.27	88.16	16.45	91.16	16.98	90.95
SSDG-R [27]		7.38	97.17	10.44	95.94	11.71	96.59	15.61	91.54	11.29	95.31
D^2AM [10]		12.70	95.66	20.98	85.58	15.43	91.22	15.27	90.87	16.10	90.83
ViT [24]		**4.75**	98.59	15.70	92.76	17.68	86.66	16.46	90.37	13.65	92.10
TransFAS [55]		7.08	96.69	9.81	96.13	10.12	95.53	15.52	91.10	10.63	94.86
DADN-CDS [58]		5.24	98.06	6.84	97.95	10.64	95.14	13.77	93.09	**9.12**	96.06
CIFAS [33]		5.95	96.32	10.66	95.30	**8.50**	**97.24**	13.17	93.44	9.57	95.58
CF-PAD [15]		8.11	96.43	11.78	95.64	16.50	91.50	9.87	95.13	11.57	94.68
	ViT-FS	11.19	96.09	9.89	95.67	14.90	93.59	**5.52**	**98.60**	10.37	95.99
ViT-B	FE	30.71	77.50	18.67	90.33	36.10	72.71	27.07	80.74	28.14	80.32
	FoundPAD (ours)	20.95	89.88	**4.89**	98.08	10.45	95.80	6.19	98.31	10.62	95.52
	ViT-FS	8.10	98.12	26.11	82.97	21.55	87.29	36.40	69.57	23.04	84.49
ViT-L	FE	21.67	86.87	9.00	96.10	22.05	86.27	22.32	84.85	18.76	88.52
	FoundPAD (ours)	16.90	93.18	6.00	**98.72**	9.90	96.07	5.87	98.41	9.67	**96.60**

Table 3. Results of triple-source cross-dataset evaluation on the CA dataset. The best and second-best results for each metric are highlighted in bold and underlined, respectively. FoundPAD surpasses both ViT-FS and FE in all considered scenarios while achieving superior performances than SOTA methods for ViT-B.

Method		O&C&M → CA	
		HTER(%) ↓	AUC(%) ↑
GRL Layer [21]		29.1	76.4
ADDA [49]		33.7	70.3
DA-FAS [44]		27.1	79.2
UCDA-FAS [40]		26.1	80.0
CIFAS [33]		24.6	83.2
CF-PAD [15]		23.5	84.2
	ViT-FS	16.0	89.1
ViT-B	FE	23.7	84.2
	FoundPAD (ours)	**15.6**	**91.0**
	ViT-FS	48.2	52.3
ViT-L	FE	43.9	58.3
	FoundPAD (ours)	43.0	59.7

Table 4. Results of double-source cross-dataset evaluation. The best and second-best results for each metric are highlighted in bold and underlined, respectively. FoundPAD surpasses both ViT-FS and FE in all the considered scenarios while achieving superior performances than SOTA PAD methods.

Method		M&I → C		M&I → O	
		HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑
MS-LBP [36]		51.16	52.09	43.63	58.07
IDA [56]		45.16	58.80	54.52	42.17
MADDG [45]		41.02	64.33	39.35	65.10
RFM [46]		36.34	67.52	29.12	72.61
SSDG-R [27]		31.89	71.29	36.01	66.88
DR-MD-Net [53]		31.67	75.23	34.02	72.65
D^2AM [10]		32.65	72.04	27.70	75.36
CIFAS [33]		22.67	83.39	24.63	81.48
CF-PAD [15]		22.11	85.06	19.71	89.01
	ViT-FS	14.00	92.97	**7.11**	**97.88**
ViT-B	FE	27.22	79.94	33.57	72.70
	FoundPAD (ours)	13.22	93.97	9.31	96.69
	ViT-FS	25.22	85.66	9.07	96.41
ViT-L	FE	11.33	94.57	26.19	81.31
	FoundPAD (ours)	**4.67**	**99.22**	10.23	95.58

nario (Table 4), FoundPAD achieved high AUCs and low HTERs for both architectures. For single-source PAD (Table 5), FoundPAD surpassed ViT-FS in 7 of the 12 evaluated scenarios with ViT-B, and in 9 of the 12 evaluated scenarios with ViT-L, while outperforming FE in all evaluated scenarios regardless of the architecture. The comparison between FoundPAD and FE showed improved performance in all the analysed scenarios for triple, double and single-source settings, showing that adapting the embedding space for PAD-specific nuances through embedding space alignment was necessary. Furthermore, FoundPAD performed very competitively to ViT-FS, highlighting the benefits of taking advantage of the built-in knowledge of the pre-trained FM.

Comparison with SOTA: As PAD research is always evolving and new approaches have been proposed rapidly in the last years, we did our best to provide the most comprehensive comparison with the recent works across all experimental setups, however, we acknowledge that the comparison might have missed specific works and acts as a tool to place the achieved performances within the scope of SOTA, rather than a comprehensive comparison. We also state that we list the works that evaluated the corresponding protocols in each table, so the approaches in each table might not completely overlap. In this part, we will only compare with our proposed method, FoundPAD. The comparison with the SOTAs for the triple-source scenario is gathered in Tables 2 and 3 as different protocols were followed by the different methods. Table 2 shows that the best average AUC is achieved by FoundPAD-ViT-L (96.60%), followed by TransFAS [55] (96.06%). For the evaluations on C and O, FoundPAD-ViT-B and FoundPAD-ViT-L share 1^{st} and 2^{nd}

Table 5. Results of single-source cross-dataset evaluation. The best and second-best results for each metric are highlighted in bold and underlined, respectively. FoundPAD surpasses both ViT-FS and FE in most of the considered scenarios while achieving superior average performances than SOTA PAD methods by a large margin.

Method		C → I	C → M	C → O	I → C	I → M	I → O	M → C	M → I	M → O	O → I	O → M	O→ C	Average	Worst
Binary CNN [59]		45.80	25.60	36.40	44.40	48.60	45.40	50.10	49.90	31.40	47.40	30.20	41.20	41.37 ± 8.42	50.10
ADA [52]		17.50	9.30	29.10	41.60	30.50	39.60	17.70	5.10	31.20	26.80	31.50	19.80	24.98 ± 11.28	41.60
DR-MD-Net [53]		26.10	20.20	24.70	39.20	23.20	33.60	34.30	8.70	31.70	27.60	22.00	21.80	26.09 ± 7.70	39.20
DR-UDA [54]		15.60	9.00	28.70	34.20	29.00	38.50	16.80	**3.00**	30.20	25.40	27.40	19.50	23.11 ± 10.50	38.50
CF-PAD [15]		24.80	17.14	19.43	34.00	24.76	31.70	14.44	15.90	25.34	21.50	15.00	20.33	22.03 ± 6.33	34.00
ViT-B	ViT-FS	26.05	8.33	17.79	24.33	22.38	18.43	**4.00**	15.05	**6.68**	20.45	15.48	11.56	15.88 ± 7.07	26.05
	FE	32.95	35.24	30.63	38.56	35.71	40.52	25.33	30.45	28.86	34.65	32.86	16.89	31.89 ± 6.29	40.52
	FoundPAD (ours)	16.40	24.52	15.14	17.00	**18.57**	**13.38**	20.00	17.10	19.41	**8.95**	23.33	7.89	16.81 ± 5.03	24.52
ViT-L	ViT-FS	22.60	**5.71**	37.07	25.00	20.48	23.62	7.89	15.00	22.93	26.05	**11.43**	20.33	19.84 ± 8.69	37.07
	FE	25.70	30.24	25.06	19.78	26.90	31.83	15.44	19.00	28.37	24.40	32.62	13.78	24.43 ± 6.21	32.62
	FoundPAD (ours)	**14.05**	21.43	**11.00**	**10.22**	19.29	16.94	12.00	14.55	20.93	14.40	23.81	**7.22**	**15.49 ± 5.07**	**23.81**

Table 6. Results of single-source cross-dataset evaluation when training on the synthetic dataset SynthASpoof for previously proposed solutions and our proposed method, FoundPAD. The best and second-best results for each metric are highlighted in bold and underlined, respectively. FoundPAD presents the best average AUC and second-best average HTER, highlighting its generalizability to unseen domains.

Method		M		C		I		O		Average	
		HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑
ResNet [18]		25.48	79.54	39.22	62.00	8.90	96.96	34.23	71.48	26.96	77.50
PixBis [18]		38.33	63.87	38.44	64.79	7.50	96.88	35.77	63.50	30.74	72.26
ViT-SIDE B [19]		36.67	69.78	33.33	75.21	9.80	96.67	**13.26**	**94.04**	**23.27**	83.93
SynFace Co-Former A [19]		18.57	90.76	41.11	64.49	16.30	92.31	21.67	86.44	24.41	83.50
SynFace Co-Former B [19]		**16.67**	**91.61**	40.00	63.05	18.80	88.20	25.35	82.02	25.21	81.22
CoDe-Lc [19]		37.14	71.45	37.11	69.08	12.10	95.31	37.58	66.30	30.98	75.54
CoDe-Lh [19]		39.05	70.58	39.33	63.70	13.90	93.84	38.11	68.33	32.60	74.11
OrthPADNet [19]		20.95	87.59	39.78	67.32	23.70	79.55	34.92	71.69	29.84	76.54
idvcVT [19]		45.71	64.58	56.44	41.82	23.10	85.08	51.09	49.68	44.09	60.29
hdaFVPAD [19]		65.71	31.95	71.33	21.95	47.80	52.10	37.89	66.49	55.68	43.12
ViT-B	ViT-FS	50.24	58.61	44.44	59.46	24.40	81.48	46.53	56.08	41.40	63.91
	FE	47.14	59.27	28.11	78.81	19.50	87.08	40.28	63.66	33.76	72.21
	FoundPAD (ours)	47.14	66.18	27.33	83.03	16.15	90.79	33.12	73.56	30.94	78.39
ViT-L	ViT-FS	50.00	55.94	47.11	58.14	33.60	73.20	50.04	50.63	45.19	59.48
	FE	52.62	55.76	13.89	92.82	20.50	87.94	29.58	77.13	29.15	78.41
	FoundPAD (ours)	45.71	69.76	**9.89**	**96.03**	**6.40**	**98.58**	32.05	75.69	23.51	**85.01**

places. Evaluation on I shows competitive results with other methods, whereas evaluation on M is where FoundPAD needs improvement. Table 3 shows that FoundPAD-ViT-B achieves improved results, whereas FoundPAD-ViT-L is not performing on the same level. For the double-source scenario (Table 4), FoundPAD-ViT-B and FoundPAD-ViT-L are the only methods that surpass the 90% AUC mark, where FoundPAD-ViT-L reaches an AUC of 99.22% on C and FoundPAD-ViT-B reaches an AUC of 96.69% on O. In the challenging low data availability scenario (Table 5), FoundPAD-ViT-B improved the best average HTER by 5.22 pp. while reducing the lowest worst achieved HTER from 34.00% to 24.52%. FoundPAD-ViT-L improved the best average HTER by 6.54 pp. while further reducing the lowest worst achieved HTER to 23.81%. This illustrates FoundPADs' ability to use the induced knowledge from the FM.

Synthetic Data Applicability for FoundPAD: We further assess FoundPAD's performance when fine-tuned with the privacy-friendly synthetic SynthASpoof [18] training dataset. The obtained results are compared with previous methods [18, 19] in Table 6. FoundPAD obtained comparable results with the winner of the SynFacePAD 2023 competition [19], ViT-SIDE B, which proposed a ViT model architecture pre-trained on ImageNet [14]. We also note that our conclusions on comparing ViT-FS, FE, and FoundPAD still stand when using synthetic data.

6. Conclusion

This work is the first to adapt FMs using LoRA to perform a PAD task within low data availability. Specifically, we adapt the pre-trained FM CLIP to the PAD task using LoRA, while simultaneously training a header to perform classification. This allows the FM to adapt its feature space to the downstream PAD task without losing the knowledge acquired during its self-supervised pre-training process. To show that FoundPAD is effectively taking advantage of FMs' properties, we further propose and evaluate three alternative FM and transformer-based methods, TI, ViT-FS and FE, which we compare with FoundPAD on different data availability settings. These experiments proved the overall superiority of FoundPAD compared with the remaining approaches, highlighting that the in-built knowledge of the FM is beneficial for the downstream task while not being enough to achieve good classification performances in domain-specific tasks such as PAD. FoundPAD achieved competitive results or even surpassed PAD SOTA methods and experiments using synthetic training data demonstrated its ability to generalize to the unseen authentic domain on several evaluation benchmarks. These outcomes show the feasibility of using FoundPAD to tackle common PAD issues due to its high generalization capacity, namely in the challenging low data availability scenario.

Acknowledgments: This research work has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

[1] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020. 4

[2] Dosovitskiy Alexey. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929*, 2020. 5

[3] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. 1

[4] Zinelabdine Boulkenafet, Jukka Komulainen, Lei Li, Xiaoyi Feng, and Abdenour Hadid. OULU-NPU: A mobile face presentation attack database with real-world variations. In *FG*, pages 612–618. IEEE Computer Society, 2017. 1, 5

[5] Fadi Boutros, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Elasticface: Elastic margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1578–1587, 2022. 1

[6] Fadi Boutros, Meiling Fang, Marcel Klemt, Biying Fu, and Naser Damer. Cr-fiqa: Face image quality assessment by learning sample relative classifiability, 2023. 5

[7] Feng Chen, Mario Valerio Giuffrida, and Sotirios A Tsaftaris. Adapting vision foundation models for plant phenotyping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 604–613, 2023. 3

[8] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 2, 3

[9] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 2, 3

[10] Zhihong Chen, Taiping Yao, Kekai Sheng, Shouhong Ding, Ying Tai, Jilin Li, Feiyue Huang, and Xinyu Jin. Generalizable representation learning for mixture domain face anti-spoofing. In *AAAI*, pages 1132–1139. AAAI Press, 2021. 2, 5, 7

[11] Tahar Chettaoui, Naser Damer, and Fadi Boutros. Froundation: Are foundation models ready for face recognition? *arXiv preprint arXiv:2410.23831*, 2024. 1, 2, 3, 4, 6

[12] Ivana Chingovska, André Anjos, and Sébastien Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *BIOSIG*, volume P-196 of *LNI*, pages 1–7. GI, 2012. 5

[13] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 1

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 8

[15] Meiling Fang and Naser Damer. Face presentation attack detection by excavating causal clues and adapting embedding statistics. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6269–6279, 2024. 1, 2, 5, 6, 7, 8

[16] Meiling Fang, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Learnable multi-level frequency decomposition and hierarchical attention mechanism for generalized face presentation attack detection. In *IEEE/CVF WACV, Waikoloa, HI, USA, January 3-8, 2022*, pages 1131–1140. IEEE, 2022. 1, 2, 5, 6, 7

[17] Meiling Fang, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Real masks and spoof faces: On the masked face presentation attack detection. *Pattern Recognit.*, 123:108398, 2022. 1

[18] Meiling Fang, Marco Huber, and Naser Damer. Synthaspoof: Developing face presentation attack detection based on privacy-friendly synthetic data. In *CVPR Workshops*, pages 1061–1070. IEEE, 2023. 1, 2, 5, 8

[19] Meiling Fang, Marco Huber, Julian Fiérrez, Raghavendra Ramachandra, Naser Damer, Alhasan Alkhaddour, Maksim Kasantcev, Vasiliy Pryadchenko, Ziyuan Yang, Huijie Huangfu, Yingyu Chen, Yi Zhang, Yuchen Pan, Junjun Jiang, Xianming Liu, Xianyun Sun, Caiyong Wang, Xingyu Liu, Zhaohua Chang, Guangzhe Zhao, Juan E. Tapia, Lázaro J. González Soler, Carlos M. Aravena, and Daniel Schulz. Synfacepad 2023: Competition on face presentation attack detection based on privacy-aware synthetic training data. In *IJCB*, pages 1–11. IEEE, 2023. 8

[20] Parisa Farmanifard and Arun Ross. Iris-sam: Iris segmentation using a foundational model. *arXiv preprint arXiv:2402.06497*, 2024. 1, 2, 3

[21] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1180–1189. JMLR.org, 2015. 7

[22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 3, 4

[23] Hsin-Ping Huang, Deqing Sun, Yaojie Liu, Wen-Sheng Chu, Taihong Xiao, Jinwei Yuan, Hartwig Adam, and Ming-Hsuan Yang. Adaptive transformers for robust few-shot cross-domain face anti-spoofing. In *ECCV (13)*, volume 13673 of *Lecture Notes in Computer Science*, pages 37–54. Springer, 2022. 5

[24] Hsin-Ping Huang, Deqing Sun, Yaojie Liu, Wen-Sheng Chu, Taihong Xiao, Jinwei Yuan, Hartwig Adam, and Ming-Hsuan Yang. Adaptive transformers for robust few-shot cross-domain face anti-spoofing. In *ECCV (13)*, volume 13673 of *Lecture Notes in Computer Science*, pages 37–54. Springer, 2022. 7

[25] International Organization for Standardization. ISO/IEC DIS 30107-3:2016: Information Technology – Biometric presentation attack detection – P. 3: Testing and reporting, 2017. 6

[26] International Organization for Standardization. ISO/IEC 2382-37:2022: Information technology — Vocabulary — Part 37: Biometrics, 2022. 3

[27] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-side domain generalization for face anti-spoofing. In *CVPR*, pages 8481–8490. Computer Vision Foundation / IEEE, 2020. 1, 2, 5, 7

[28] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023. 4

[29] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020. 5

[30] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1, 2, 3

[31] Haoliang Li, Wen Li, Hong Cao, Shiqi Wang, Feiyue Huang, and Alex C. Kot. Unsupervised domain adaptation for face anti-spoofing. *IEEE Trans. Inf. Forensics Secur.*, 13(7):1794–1809, 2018. 2

[32] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *CVPR*, pages 5400–5409. IEEE Computer Society, 2018. 5, 6

[33] Yuchen Liu, Yabo Chen, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Causal intervention for generalizable face anti-spoofing. In *ICME*, pages 1–6. IEEE, 2022. 2, 5, 6, 7

[34] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *CVPR*, pages 389–398. IEEE Computer Society, 2018. 1, 2, 7

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[36] Jukka Määttä, Abdenour Hadid, and Matti Pietikäinen. Face spoofing detection from single images using micro-texture analysis. In *IJCB*, pages 1–7. IEEE Computer Society, 2011. 7

[37] Andrey Makrushin, Andreas Uhl, and Jana Dittmann. A survey on synthetic biometrics: Fingerprint, face, iris and vascular patterns. *IEEE Access*, 11:33887–33899, 2023. 1

[38] Zuheng Ming, Zitong Yu, Musab Al-Ghadi, Muriel Visani, Muhammad Muzzamil Luqman, and Jean-Christophe Burie. Vitranspad: Video transformer using convolution and self-attention for face presentation attack detection. In *ICIP*, pages 4248–4252. IEEE, 2022. 5, 7

[39] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3

[40] Ankush Panwar, Pratyush Singh, Suman Saha, Danda Pani Paudel, and Luc Van Gool. Unsupervised compound domain adaptation for face anti-spoofing. In *FG*, pages 1–8. IEEE, 2021. 5, 7

[41] Foivos Paraperas Papantoniou, Alexandros Lattas, Stylianos Moschoglou, Jiankang Deng, Bernhard Kainz, and Stefanos Zafeiriou. Arc2face: A foundation model for id-consistent human faces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 1, 2, 3

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 4, 5

[43] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1

[44] Suman Saha, Wenhao Xu, Menelaos Kanakis, Stamatios Georgoulis, Yuhua Chen, Danda Pani Paudel, and Luc Van Gool. Domain agnostic feature learning for image and video based face anti-spoofing. In *CVPR Workshops*, pages 3490–3499. IEEE, 2020. 7

[45] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C. Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *CVPR*, pages 10023–10031. Computer Vision Foundation / IEEE, 2019. 1, 2, 5, 6, 7

[46] Rui Shao, Xiangyuan Lan, and Pong C. Yuen. Regularized fine-grained meta face anti-spoofing. In *AAAI*, pages 11974–11981. AAAI Press, 2020. 1, 2, 5, 6, 7

[47] Koushik Srivatsan, Muzammal Naseer, and Karthik Nandakumar. FLIP: cross-domain face anti-spoofing with language guidance. In *ICCV*, pages 19628–19639. IEEE, 2023. 1, 2

[48] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *Advances in neural information processing systems*, 32, 2019. 6

[49] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971. IEEE Computer Society, 2017. 7

[50] An Wang, Mobarakol Islam, Mengya Xu, Yang Zhang, and Hongliang Ren. SAM meets robotic surgery: An empirical

study on generalization, robustness and adaptation. *CoRR*, abs/2308.07156, 2023. 2

[51] Chien-Yi Wang, Yu-Ding Lu, Shang-Ta Yang, and Shang-Hong Lai. Patchnet: A simple face anti-spoofing framework via fine-grained patch recognition. In *CVPR*, pages 20249–20258. IEEE, 2022. 7

[52] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Improving cross-database face presentation attack detection via adversarial domain adaptation. In *ICB*, pages 1–8. IEEE, 2019. 2, 8

[53] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *CVPR*, pages 6677–6686. Computer Vision Foundation / IEEE, 2020. 5, 7, 8

[54] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Unsupervised adversarial domain adaptation for cross-domain face presentation attack detection. *IEEE Trans. Inf. Forensics Secur.*, 16:56–69, 2021. 5, 8

[55] Zhuo Wang, Qiangchang Wang, Weihong Deng, and Guodong Guo. Face anti-spoofing using transformers with relation-aware mechanism. *IEEE Trans. Biom. Behav. Identity Sci.*, 4(3):439–450, 2022. 1, 2, 7

[56] Di Wen, Hu Han, and Anil K. Jain. Face spoof detection with image distortion analysis. *IEEE Trans. Inf. Forensics Secur.*, 10(4):746–761, 2015. 5, 7

[57] Zheng Xu, Wen Li, Li Niu, and Dong Xu. Exploiting low-rank structure from latent domains for domain generalization. In *ECCV (3)*, volume 8691 of *Lecture Notes in Computer Science*, pages 628–643. Springer, 2014. 7

[58] Wenjun Yan, Ying Zeng, and Haifeng Hu. Domain adversarial disentanglement network with cross-domain synthesis for generalized face anti-spoofing. *IEEE Trans. Circuits Syst. Video Technol.*, 32(10):7033–7046, 2022. 1, 2, 7

[59] Jianwei Yang, Zhen Lei, and Stan Z. Li. Learn convolutional neural network for face anti-spoofing. *CoRR*, abs/1408.5601, 2014. 6, 8

[60] Zitong Yu, Xiaobai Li, Jingang Shi, Zhaoqiang Xia, and Guoying Zhao. Revisiting pixel-wise supervision for face anti-spoofing. *IEEE Trans. Biom. Behav. Identity Sci.*, 3(3):285–295, 2021. 1, 2, 5, 7

[61] Zitong Yu, Jun Wan, Yunxiao Qin, Xiaobai Li, Stan Z. Li, and Guoying Zhao. NAS-FAS: static-dynamic central difference network search for face anti-spoofing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(9):3005–3023, 2021. 1, 2, 5, 7

[62] Bowen Zhang, Ying Chen, Long Bai, Yan Zhao, Yuxiang Sun, Yixuan Yuan, Jianhua Zhang, and Hongliang Ren. Learning to adapt foundation model dinov2 for capsule endoscopy diagnosis. *arXiv preprint arXiv:2406.10508*, 2024. 3

[63] Haoyu Zhang, Raghavendra Ramachandra, Kiran Raja, and Christoph Busch. Generalized single-image-based morphing attack detection using deep representations from vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1510–1518, June 2024. 5

[64] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 5

[65] Yuanhan Zhang, Zhenfei Yin, Yidong Li, Guojun Yin, Junjie Yan, Jing Shao, and Ziwei Liu. Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations. In *ECCV. Glasgow, UK, August 23-28, 2020,*, volume 12357 of *Lecture Notes in Computer Science*, pages 70–85. Springer, 2020. 1, 2, 5

[66] Zhiwei Zhang, Junjie Yan, Sifei Liu, Zhen Lei, Dong Yi, and Stan Z. Li. A face antispoofing database with diverse attacks. In *ICB*, pages 26–31. IEEE, 2012. 1, 5