Spoof Trace Disentanglement for Generic Face Anti-Spoofing

Yaojie Liu, and Xiaoming Liu, *Member, IEEE*

Abstract—Prior studies show that the key to face anti-spoofing lies in the subtle image patterns, termed "spoof trace", *e.g.*, color distortion, 3D mask edge, and Moiré pattern. Spoof detection rooted on those spoof traces can improve not only the model's generalization but also the interpretability. Yet, it is a challenging task due to the diversity of spoof attacks and the lack of ground truth for spoof traces. In this work, we propose a novel adversarial learning framework to explicitly estimate the spoof related patterns for face anti-spoofing. Inspired by the physical process, spoof faces are disentangled into spoof traces and the live counterparts in two steps: additive step and inpainting step. This two-step modeling can effectively narrow down the searching space for adversarial learning of spoof trace. Based on the trace modeling, the disentangled spoof traces can be utilized to reversely construct new spoof faces, which is used as data augmentation to effectively tackle long-tail spoof types. In addition, we apply frequency-based image decomposition in both the input and disentangled traces to better reflect the low-level vision cues. Our approach demonstrates superior spoof detection performance on 3 testing scenarios: known attacks, unknown attacks, and open-set attacks. Meanwhile, it provides a visually-convincing estimation of the spoof traces. Source code and pre-trained models will be publicly available upon publication.

Index Terms—Face Anti-Spoofing, Low-level Vision, Weak Supervision, Synthesis, Spoof Traces, Deep Learning.

1 Introduction

In recent years, the vulnerability of face biometric systems has been widely recognized and increasingly brought attention to the computer vision community. The attacks attempt to deceive the systems to make wrong identity recognition: either recognize the attackers as a target person (*i.e.*, impersonation), or cover up the original identity (*i.e.*, obfuscation). There are various types of digital and physical attacks, including face morphing [1]–[3], face adversarial attacks [4]–[6], face manipulation attacks (*e.g.*, deepfake, face swap) [7], [8], and face spoofing [9]–[11]. Among all, face spoofing is the only physical attack, where attackers present faces from spoof mediums without the need to know any details of the machine learning models. Typical spoof mediums include photograph, screen, mask and makeup. These spoof mediums can be easily manufactured by ordinary people, and hence pose huge threats to face biometric applications such as mobile face unlock, building access control, and transportation security. Therefore, face biometric systems need to be secured with face anti-spoofing (FAS) techniques to distinguish the source of the face before performing the face recognition task.

Due to the hardware dependency on face recognition systems, most FAS studies are based on monocular RGB camera for these two decades. One of the most common approaches is based on texture analysis [12]–[14]. Researchers notice that presenting faces from spoof mediums introduces special texture differences, such as color distortions, unnatural specular highlights, Moiré patterns, *etc*. Those texture differences are inherent with spoofing process and thus hard to remove or camouflage. Conventional approaches build a feature extractor plus classifier pipeline, such as local binary patterns+support vector machine or histogram of oriented gradients+support vector machine [15], [16]. Those methods perform well on several small-scale databases with controlled environments. In recent years, many works leverage deep learning techniques and show great progress in more various environment [17]–[21]. Deep learning based methods can be generally grouped into 3 categories: direct FAS, auxiliary FAS, and generative FAS, as illustrated in Fig. 2. Early works [22], [23] build vanilla CNN with binary output to directly predict the spoofness of an input face (Fig.2a). Methods [18], [21] propose to

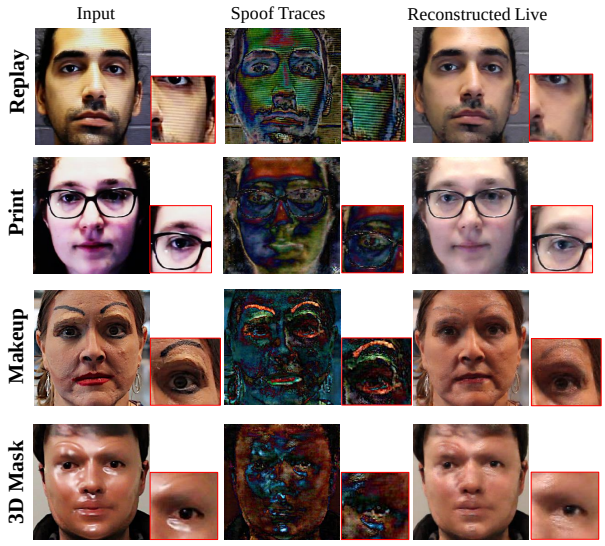
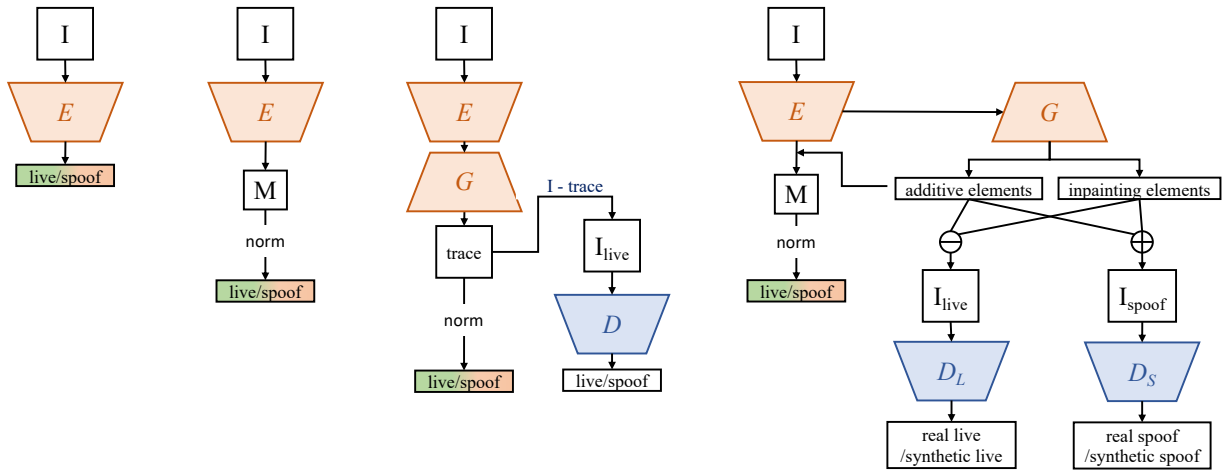


Fig. 1: The proposed approach can detect spoof faces, disentangle the spoof traces, and reconstruct the live counterparts. It can be applied to diverse spoof types and estimate distinct traces (*e.g.*, Moiré pattern in replay attack, artificial eyebrow and wax in makeup attack, color distortion in print attack, and specular highlights in 3D mask attack). Zoom in for details.

• *Y. Liu and X. Liu are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA. Email: {liuyaoj1,liuxm}@msu.edu.*



(a) direct FAS (b) auxiliary FAS (c) generative FAS (d) proposed method

Fig. 2: The comparison of different deep-learning based face anti-spoofing. (a) direct FAS only provides a binary decision of spoofness; (b) auxiliary FAS can provide simple interpretation of spoofness. M denotes the auxiliary task, such as depth map estimation; (c) generative FAS can provide more intuitive interpretation of spoofness, but only for a limited number of spoof attacks; (d) the proposed method can provide spoof trace estimation for generic face spoof attacks.

learn an intermediate representation, *e.g.*, depth, rPPG, reflection, instead of binary classes, which can lead to better generalization and performance with limited data (Fig.2b). [24]–[26] additionally attempt to generate the visual patterns existing in the spoof samples (Fig.2c), providing a more intuitive interpretation of the sample's spoofness.

Despite the success, there are three unsolved challenges in tackling FAS via deep learning. First, most prior works design FAS for only few spoof types, either print/replay or 3D mask solely. In practice, FAS would encounter a wide variety of spoof types including print, replay, 3D masks, facial makeup, and even known types. Therefore, FAS models need to be designed to address different attack properties, and be evaluated on both known attacks and unknown attacks to better reflect real-world performance. Second, unlike many classification tasks that mostly rely on high-level vision cues, FAS largely rely on texture(*i.e.*, the low-level vision cues), which is not straightforward even for human vision. Formulating FAS as a binary classification problem would lead to overfitting and poor interpretability. Although auxiliary FAS and generative FAS attempt to offer some extent of interpretation by fixation, saliency, or noise analysis, there is little understanding on what the exact differences are between live and spoof, and what patterns the classifier's decision is based upon. Third, compared with other face related tasks such as recognition or alignment, FAS data has several limitations. Most FAS data are manually collected in the controlled indoor environment, which has limited inter-subject variation and environment variation. Moreover, some spoof types are hard to have large-scale collection. For example, facial makeup attacks require artists with special expertise, and each attack takes a very long time to prepare. Customized silicone mask also requires few specific companies to make, and it often has high cost. It would results in a long-tail situation for those attack types, creating extra difficulties for FAS model.

In this work, we aim to explore the exact patterns differentiating a spoof face and its live counterpart, termed as **spoof trace**. We equip this model with the ability to explicitly disentangle the spoof traces from the input faces and make spoof classification based on the spoof traces. Some examples of spoof traces are shown in Fig. 1. As seen in the figure, our model considers a wide variety of spoof types as a **generic FAS** solution. This is a challenging objective due to the diversity of spoof traces and the lack of ground truth during model learning. However, we believe fulfilling this objective can bring several benefits:

1) Binary FAS models would harvest any cue that helps classification, which might include spoof-irrelevant cues such as lighting, and thus hinder generalization. In contrast, spoof trace disentanglement explicitly tackles the most fundamental cue in spoofing, upon which the classification can be more grounded and witness better generalization.
2) With the trend of pursuing explainable AI [27], [28], it is desirable for FAS model to provide explanation to support its decision. Spoof trace serves as a good visual explanation to the spoof classification. Certain properties (*e.g.*, severity, methodology) of spoof attacks might potentially be revealed from the traces.
3) Well disentangled spoof traces can be used for synthesizing new spoof samples, which can mitigate the issue of long-tail spoof data, such as special 3D masks and makeup.

Shown in Fig. 2d, we propose a novel Spoof Trace Disentanglement Network (STDN+) to explore the spoof traces for generic face anti-spoofing. The preliminary version of STDN [29] assume an additive process for disentangling the spoof traces and regularize the process with Frobenius norm. This formulation penalizes large trace intensity to prevent identity shift but also lead to sub-optimal results for attacks with significant appearance changes. In STDN+, we design a two-step process to disentangle the input face into spoof traces and the live counterparts. The first step considers patterns to be additive and the live counterpart can be recovered by removing them similar to a de-noising process. The second step continue updating the area-of-interest from the first step as an inpainting process [30], [31], without any regularization on the intensity changes. This two-step modeling can provide a better regularization and narrow down the searching space for adversarial learning of spoof trace. In addition, as STDN [29] demonstrates the effectiveness of multiscale designs in handling texture with different frequency properties, we extend such designs in STDN+ to input

images, traces, and adversarial learning. With only binary label as weak supervision, we adopt an overall generative adversarial network (GAN) based training strategy to learn the spoof traces. We apply an encoder-decoder to disentangle the input face into spoof trace and the live counterpart. Spoof trace then can be used to synthesize new spoof samples with other live faces. The live counterparts and synthesized spoof are sent to discriminators for adversarial training. A depth map network takes the encoder feature and spoof traces as input and then estimate a pseudo-depth map for spoof classification. The synthesized spoof are further utilized to train the spoof disentangling in a fully supervised fashion, thanks to disentangled spoof traces as ground truth for the synthesized data.

A preliminary version of this work was published in the Proceedings European Conference on Computer Vision (ECCV) 2020 [29]. We extend the work from three aspects. First, we reformulate the disentangling process as a two-step process. The two-step process can provide a more proper regularization on spoof trace learning for various attack types. Significant improvements are shown in several spoof types, such as paper glasses and 3D masks. Second, we observe the effectiveness on multi-scale designs in STDN (*i.e.*, multi-scale traces and discriminators) and extend the multi-scale designs to input to better extract low-level texture information. Third, we provide a more comprehensive evaluation and comparison of STDN+ performance, including open-set testing scenario, comparison with other spoof formulations, and more results and analysis.

The main contributions of this work are as follows:

- We for the first time study spoof trace for *generic* face anti-spoofing, where a wide variety of spoof types are tackled with one unified framework;
- We propose a novel physics-guided model to disentangle spoof traces, and utilize the spoof traces to synthesize new data samples for enhanced training;
- We propose novel protocols for a generic open-set face anti-spoofing;
- We achieve SOTA anti-spoofing performance and provide convincing visualization for a wide variety of spoof types.

2 Related Work

Face Anti-Spoofing Face anti-spoofing has been studied for more than a decade and its development can be roughly divided into three stages. In the early years, researchers leverage spontaneous human movement, such as eye blinking and head motion, to detect simple print photograph or static replay attacks [32], [33]. However, when facing counter attacks, such as print face with eye region cut, and replaying a face video, those methods would fail. In the second stage, researchers pay more attention to texture differences between live and spoof, which are inherent to spoof mediums. Researchers mainly extract hand-crafted features from the faces, *e.g.*, LBP [12], [15], [34], [35], HoG [16], [36], SIFT [14] and SURF [13], and train a classifier to split the live *vs.* spoof, *e.g.*, SVM and LDA.

Recently, face anti-spoofing solutions equipped with deep learning techniques have demonstrated significant improvements over the conventional methods. Methods in [22], [37]–[39] train a deep neural network to learn a binary classification between live and spoof. In [17]–[21], additional supervisions, such as face depth map and rPPG signal, are utilized to help the network to learn more generalizable features. As the latest approaches achieving saturated performance on several benchmarks, researchers start to explore more challenging cases, such as few-shot/zero-shot face anti-spoofing [19], [40], [41] and domain adaptation in face anti-spoofing [20], [42], [43].

In this work, we aim to solve an interesting yet very challenging problem: disentangling and visualizing the spoof traces from an input face. A related work [24] also adopts GAN seeking to estimate the spoof traces. However, they formulate the traces as low-intensity noises, which is limited to print and replay attacks only and cannot provide convincing visual results. In contrast, we explore spoof traces for a much wider range of spoof attacks, visualize them with novel disentanglement, and also evaluate the proposed method on the challenging cases, *e.g.*, zero-shot face anti-spoofing.

Disentanglement Learning Disentanglement learning is often adopted to better represent complex data and features. DR-GAN [44] disentangles a face into identity and pose vectors for pose-invariant face recognition and view synthesis. Similarly in gait recognition, [45] disentangles the representations of appearance, canonical, and pose features from an input gait video. 3D reconstruction works [46], [47] also disentangle the representation of a 3D face into identity, expressions, poses, albedo, and illuminations. For image synthesis, [48] disentangles an image into appearance and shape with U-Net and Variational Auto Encoder (VAE).

Different from [44]–[46], we intend to disentangle features that have different scales and contain geometric information. We leverage the multiple outputs to represent features at different scales, and adopt multiple-scale discriminators to properly learn them. Moreover, we propose a novel warping layer to tackle the geometric discrepancy during the disentanglement and reconstruction.

Image Trace Modeling Image traces are certain signals existing in the image that can reveal information about the capturing camera, imaging setting, environment, and so on. Those signals often have much lower energy compared to the image content, which needs proper modeling to explore them. [49]–[51] observe the difference of image noises, and use them to recognize the capture cameras. From the frequency domain, [25] shows the image noises from different cameras obey different noise distributions. Such techniques are applied to the field of image forensics, and later [52], [53] propose methods to remove such traces for image anti-forensics.

Recently, image trace modeling is widely used in image forgery detection and image adversarial attack detection [54], [55]. In this work, we attempt to explore the traces of spoof face presentation. Due to different spoof mediums, spoof traces show large variations in content, intensity, and frequency distribution. We propose to disentangle the traces as additive traces and inpainting trace. And for additive traces, we further decompose them based on different frequency bands.

3 Spoof Trace Disentanglement

3.1 Problem Formulation

Let the domain of live faces be denoted as $\mathcal{L} \subset \mathbb{R}^{N\times N\times 3}$ and spoof faces as $\mathcal{S} \subset \mathbb{R}^{N\times N\times 3}$, where N is the image size. We intend to estimate the spoof trace from the input faces and leverage it to obtain the correct prediction of live and spoof. To prevent network memorizing training data and switching a spoof input to a totally different live subject (*i.e.*, identity shift), our preliminary version [29] aims to find a minimum change that transfers an input face to the live domain:

$$\arg\min_{\hat{\mathbf{I}}} \|\mathbf{I} - \hat{\mathbf{I}}\|_F \; s.t. \; \mathbf{I} \in (\mathcal{S} \cup \mathcal{L}) \text{ and } \hat{\mathbf{I}} \in \mathcal{L}. \quad (1)$$

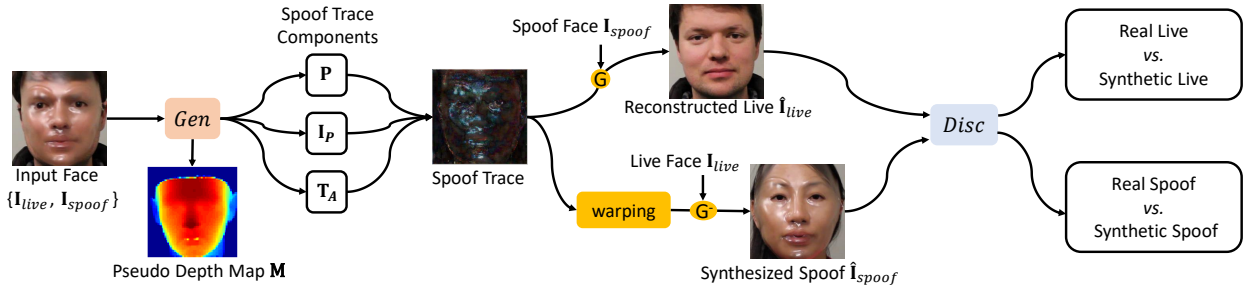


Fig. 3: The overall training workflow of Spoof Trace Disentanglement Network (STDN+). *Gen* denotes the Generator, and it takes the input faces, estimate the pseudo depth map $\mathbf{M}$, and disentangle the spoof trace components $\{\mathbf{P}, \mathbf{I}_P, \mathbf{T}_A\}$. With the spoof trace components, live reconstruction $G(\cdot)$ and spoof synthesis $G^-(\cdot)$ can be done based on Eqn. 5 and Eqn. 9. *Disc* denotes the Multi-scale Discriminators.

To represent the spoof trace, [29] considers spoofing as image degradation. Similar to many previous de-noising works [24], additive model has proven to be a proper modeling:

$$\mathbf{I}_{live} = \mathbf{I}_{spoof} - \mathbf{T}_A, \tag{2}$$

where $\mathbf{T}_A \in \mathbb{R}^{N\times N\times 3}$ denotes the additive traces. Based on this formulation, Eqn.1 becomes finding the minimum intensity change of spoof traces:

$$\arg\min_{\mathbf{T}_A} \|\mathbf{T}_A\|_F \; s.t. \; \mathbf{I} \in (\mathcal{S} \cup \mathcal{L}) \text{ and } \hat{\mathbf{I}} \in \mathcal{L}. \tag{3}$$

While considering a wide range of spoof attacks, we find Eqn.3 reveals its limitation. Some spoof attacks, such as print and replay, are physically a double-imaging process, which can be well modeled by the above formulation. Some spoof attacks, such as spoof glasses and masks, introduce large appearance discrepancy from the original skin in local regions. Applying Eqn.11 to model the bona fide content in those spoof regions would lead to sub-optimal solutions where spoof traces are still observable in the live counterpart. In contrast, disentangling those spoof attacks can be better modeled as an inpainting process:

$$\mathbf{I}_{live} = (1-\mathbf{P})\mathbf{I}_{spoof} + \mathbf{I}_P, \tag{4}$$

where $\mathbf{I}_P \in \mathbb{R}^{N\times N\times 3}$ indicates the bona fide content from spoof regions, and $\mathbf{P} \in \mathbb{R}^{N\times N\times 1}$ denotes the inpainting region. Therefore, a better spoof disentangling formulation to fit all spoof types would be a joint modeling of additive process and inpainting process as:

$$\mathbf{I}_{live} = (1-\mathbf{P})(\mathbf{I}_{spoof} - \mathbf{T}_A) + \mathbf{P}\cdot\mathbf{I}_P. \tag{5}$$

Accordingly, Eqn. 1 is re-formulated by replacing $\hat{\mathbf{I}}$ with Eqn. 5:

$$\begin{aligned} &\arg\min_{\mathbf{T}_A,\mathbf{P},\mathbf{I}_P} \|\mathbf{I} - (1-\mathbf{P})(\mathbf{I}-\mathbf{T}_A) - \mathbf{P}\cdot\mathbf{I}_P\|_F \\ \rightarrow &\arg\min_{\mathbf{T}_A,\mathbf{P},\mathbf{I}_P} \|(1-\mathbf{P})\mathbf{T}_A\|_F + \|\mathbf{P}\cdot(\mathbf{I}-\mathbf{I}_P)\|_F. \end{aligned} \tag{6}$$

As we do not wish to impose any intensity constraint on the inpainting content $\mathbf{I}_P$, the final objective becomes:

$$\arg\min_{\mathbf{T}_A,\mathbf{P}} \|(1-\mathbf{P})\mathbf{T}_A\|_F + \lambda\|\mathbf{P}\|_F \; s.t. \; \mathbf{I}\in\mathcal{S}\cup\mathcal{L}, \hat{\mathbf{I}}\in\mathcal{L}, \tag{7}$$

where λ is a weight to balance two terms. Compared with Eqn.3, Eqn.7 relaxes the intensity constraint of certain region while regularizing the total area of inpainting region.

Given a live face, one may synthesize a spoof face by inverting Eqn. 5:

$$\begin{aligned} (1-\mathbf{P})\mathbf{I}_{spoof} &= \mathbf{I}_{live} + (1-\mathbf{P})\mathbf{T}_A - \mathbf{P}\cdot\mathbf{I}_P \\ (1-\mathbf{P})\mathbf{I}_{spoof} &= (1-\mathbf{P})(\mathbf{I}_{live}-\mathbf{T}_A) + \mathbf{P}(\mathbf{I}_{live}-\mathbf{I}_P) \\ \mathbf{I}_{spoof} &= (1-\mathbf{P})(\mathbf{I}_{live}-\mathbf{T}_A) + \mathbf{P}(\mathbf{I}_{live}-\mathbf{I}_P+\mathbf{I}_{spoof}). \end{aligned} \tag{8}$$

As $\mathbf{I}_P$ attempts to approximate $\mathbf{I}_{live}$, $\mathbf{I}_{live} - \mathbf{I}_P$ can be regarded as 0. In practice, $\{\mathbf{T}_A, \mathbf{P}, \mathbf{I}_{spoof}\}$ on the right hand side come from an existing spoof sample $\mathbf{I}^i$. Therefore the spoof synthesis from $\mathbf{I}^i$ to live sample $\mathbf{I}^j$ can be represented as:

$$\hat{\mathbf{I}}^{i\rightarrow j}_{spoof} = (1-\mathbf{P}^i)(\mathbf{I}^j + \mathbf{T}^i_A) + \mathbf{P}^i\cdot\mathbf{I}^i. \tag{9}$$

We define $G(\cdot)$ to represent the reconstruction process of Eqn. 5 and $G^-(\cdot)$ to represent the synthesis of Eqn. 9.

Shown above, obtaining $\{\mathbf{T}_A, \mathbf{P}, \mathbf{I}_P\}$ from an input face $\mathbf{I}$ becomes the main goal of spoof trace disentanglement. Given that no ground truth of traces is available, we fulfill the disentanglement via generative adversarial network (GAN) based training. Shown in Fig. 3, the new Spoof Trace Disentanglement Network (STDN+) consists of a generator and discriminator. Given an input image, the generator is designed to estimate the spoof traces $\{\mathbf{T}_A, \mathbf{P}, \mathbf{I}_P\}$ and predict the spoofness (represented by the pseudo depth map) based on the traces. With the traces, we can apply function $G(\cdot)$ to reconstruct the live counterpart and function $G^-(\cdot)$ to synthesize new spoof faces. We adopt a set of discriminators at multiple image resolutions to distinguish the real faces $\{\mathbf{I}_{live}, \mathbf{I}_{spoof}\}$ with the synthetic faces $\{\hat{\mathbf{I}}_{live}, \hat{\mathbf{I}}_{spoof}\}$. We also extend our image decomposition strategy proposed in the preliminary version [29] to both generator input and additive trace output.

In the rest of this section, we present the details of generator, image decomposition, face reconstruction and synthesis, discriminator, losses, and training steps used in STDN+.

3.2 Generator

As shown in Fig. 4, the generator consists of a backbone encoder, a spoof trace decoder and a depth estimation network. The backbone encoder aims to extract multi-scale features, the depth estimation network leverages the features to estimate the facial depth map $\mathbf{M}$, and a spoof trace decoder to estimate the additive trace components $\mathbf{T}_A = \{\mathbf{B}, \mathbf{C}, \mathbf{T}\}$ and the inpainting components $\{\mathbf{P}, \mathbf{I}_P\}$. The depth map and the spoof traces will be used to compute the final spoofness score.

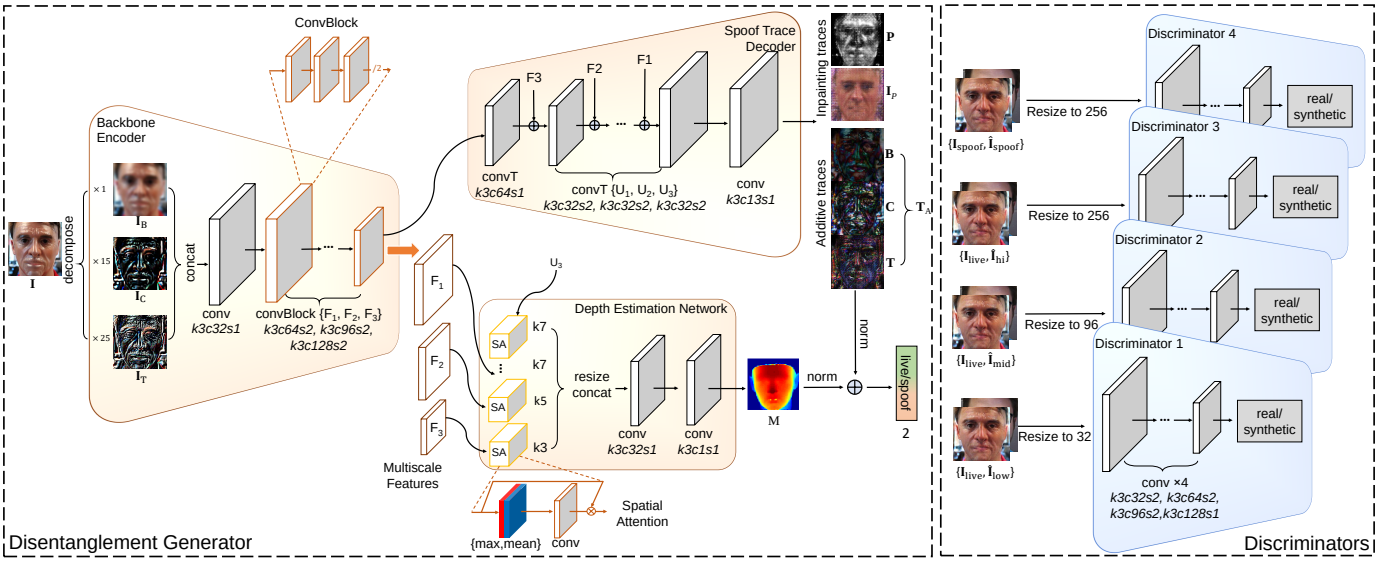


Fig. 4: The proposed STDN+ network architecture. Except the last layer, each conv and transposed conv is concatenated with a batch normalizstion layer and a leaky ReLU layer. *'k3c64s2'* indicates the kernel size of 3×3, the convolution channel of 64 and the stride of 2.

3.2.1 Image Decomposition

[24] points out that a significant amount of spoof traces, such as Moiré patterns and mask edges, exist in the high-frequency domain. However, high-frequency information often has much less energy compared with low-frequency information. An illustration is shown in Fig. 5. In the figure, we decompose the face images into 3 frequency bands: high, mid, low. The content in high- and mid- frequency bands is only visible when we multiply it 25 and 15. When handling traces from different frequency domains together, CNN might overlook the high-frequency traces as them are buried by lower-frequency ones with much stronger energy. To encourage the network to equally regard traces with different frequency properties, we explicitly decompose the input image into low-frequency element $\mathbf{I_B}$, mid-frequency element $\mathbf{I_C}$, and high-frequency element $\mathbf{I_T}$ before sending into the generator:

$$\begin{aligned} \mathbf{I_B} &= \lfloor \mathbf{I} \rfloor_{n_1}, \\ \mathbf{I_C} &= \lfloor \mathbf{I} \rfloor_{n_2} - \lfloor \mathbf{I} \rfloor_{n_1}, \\ \mathbf{I_T} &= \mathbf{I} - \lfloor \mathbf{I} \rfloor_{n2}, \end{aligned} \tag{10}$$

where $\lfloor \cdot \rfloor$ is the low bandpass filtering operation, and it's implemented by downsampling the image and upsampling it back via bilinear interpolation. n_1, n_2 are the downsampling size. We amplify the value in $\mathbf{I_C}, \mathbf{I_T}$ by 15 and 25, and then feed the concatenation of three elements to the generator. Shown in Fig. 5, traces that are less distinct in the original images become more highlighted in the $\mathbf{I_T}$ component: 3D mask and replay attack owns unique patterns different with the live face pattern, and print attack lacks necessary high frequency details.

Followed our preliminary version [29], we also decompose the additive trace $\mathbf{T}_A$ from the generator output into low-frequency traces $\mathbf{B}$, mid-frequency $\mathbf{C}$, and high-frequency $\mathbf{T}$:

$$\mathbf{T}_A = \lfloor \mathbf{B} \rfloor_{n_1} + \lfloor \mathbf{C} \rfloor_{n_2} + \mathbf{T}. \tag{11}$$

With the input size as 256, we empirically choose n_1 as 128 and n_2 as 64 to perform the decomposition.

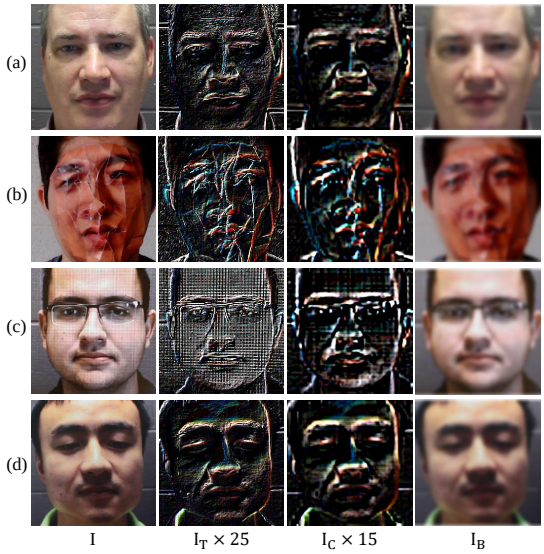


Fig. 5: The visualization of image decomposition for different input faces: (a) live face (b) 3D mask attack (c) replay attack (d) print attack.

3.2.2 Networks Architecture

Backbone encoder Backbone encoder extracts features from the input images for both depth map estimation and spoof trace disentanglement. The encoder progressively downsamples the decomposed image components 3 times to obtain features $\mathbf{F}_1 \in \mathbb{R}^{128^2 \times 64}$, $\mathbf{F}_2 \in \mathbb{R}^{64^2 \times 96}$, $\mathbf{F}_3 \in \mathbb{R}^{32^2 \times 128}$ via conv layers.

Spoof trace decoder The decoder upsamples the feature $\mathbf{F}_3$ with transpose conv layers back to the input face size 256. The last layer outputs both additive traces $\mathbf{T}_A = \{\mathbf{B}, \mathbf{C}, \mathbf{T}\}$ and inpainting components $\{\mathbf{P}, \mathbf{I}_P\}$. Similar to U-Net [56], we apply the short-cut connection between the backbone encoder and decoder to bypass the multiple scale details for a high-quality trace estimation.

Depth estimation network We still recognize the importance of the discriminative supervision used in auxiliary FAS, and thus introduce a depth estimation network to perform the pseudo-depth estimation for face anti-spoofing, as proposed in [18]. The depth

estimation network takes the concatenated features of $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{F}_3$ from the backbone encoder and $\mathbf{U}_3$ from the decoder as input. The features are put through a spatial attention mechanism from [57] and resize to the same size of $K = 32$. It outputs a face depth map $\mathbf{M} \in \mathbb{R}^{32\times32}$, where the depth values are normalized within $[0, 1]$. Regarding the number of parameters, both spoof trace decoder and depth estimation network are light weighed, while the backbone network is much heavier. With more network layers being shared to tackle both depth estimation and spoof trace disentanglement, the knowledge learnt from spoof trace disentanglement can be better shared with depth estimation task, which can lead to a better anti-spoofing performance.

3.3 Reconstruction and Synthesis

There are multiple options to use the disentangled spoof traces: 1) live reconstruction, 2) spoof synthesis, and 3) "harder" sample synthesis, which will be described below respectively.

Live reconstruction: Based on Eqn. 5, we propose a hierarchical reconstruction of the live face counterpart from the input images. To reconstruct faces at a certain resolution, each additive trace is included only if its frequency domain is lower than the target resolution. We apply $\{hi, mid, low\}$ three resolution settings as:

$$\begin{aligned}\hat{\mathbf{I}}_{hi} &= (1-\mathbf{P})(\mathbf{I} - \lfloor\mathbf{B}\rfloor_{n_1} - \lfloor\mathbf{C}\rfloor_{n_2} - \mathbf{T}) + \mathbf{P}\cdot\mathbf{I}_P,\\ \hat{\mathbf{I}}_{mid} &= (1-\mathbf{P})(\lfloor\mathbf{I}\rfloor_{n_2} - \lfloor\mathbf{B}\rfloor_{n_1} - \lfloor\mathbf{C}\rfloor_{n_2}) + \mathbf{P}\cdot\mathbf{I}_P,\\ \hat{\mathbf{I}}_{low} &= (1-\mathbf{P})(\lfloor\mathbf{I}\rfloor_{n_1} - \lfloor\mathbf{B}\rfloor_{n_1}) + \mathbf{P}\cdot\mathbf{I}_P.\end{aligned} \tag{12}$$

Spoof synthesis: Based on Eqn. 9, we can obtain a new spoof face via applying the spoof traces disentangled from a spoof face $\mathbf{I}_i$ to a live face $\mathbf{I}_j$. However, spoof traces may contain face-dependent content associated with the original spoof subject. Directly applying them to a new face with different shapes or poses may result in misalignment and strong visual implausibility. Therefore, the spoof trace should go through a geometry correction before performing this synthesis. We propose an online 3D warping layer and will introduce it in the following subsection.

"Harder" sample synthesis: The disentangled spoof traces can not only reconstruct live and synthesize new spoof, but also synthesize "harder" spoof samples by removing or amplifying part of the spoof traces. We can tune one or some of the trace elements $\{\mathbf{B}, \mathbf{C}, \mathbf{T}, \mathbf{P}\}$ to make the spoof sample to become "less spoofed", which is thus closer to a live face since the spoof traces are weakened. Such spoof data can be regarded as *harder* samples and may benefit the generalization of the disentanglement generator. For instance, while removing the low frequency element $\mathbf{B}$ from a replay spoof trace, the generator may be forced to rely on other elements such as high-level texture patterns. To synthesize the "harder" sample $\hat{\mathbf{I}}_{hard}$, we follow Eqn. 9 with two minor changes: 1) generate 3 random weights between $[0, 1]$ and multiple each with one component of $\{\mathbf{B}, \mathbf{C}, \mathbf{T}\}$; 2) randomly remove the inpainting process (*i.e.*, set $\mathbf{P} = 0$) with a probability of 0.5. Compared with other methods, such as brightness and contrast change [58], reflection and blurriness effect [21], or 3D distortion [59], our approach can introduce more realistic and effective data samples, as shown in Sec. 4.

3.3.1 Online 3D Warping Layer

We propose an online 3D warping layer to correct the shape discrepancy. To obtain the warping, previous methods in [18], [60] use offline face swapping and pre-computed dense offset respectively, where both methods are non-differentiable as well as memory intensive. In contrast, our warping layer is designed to be both differentiable and computationally efficient, which is necessary for online synthesis during the training.

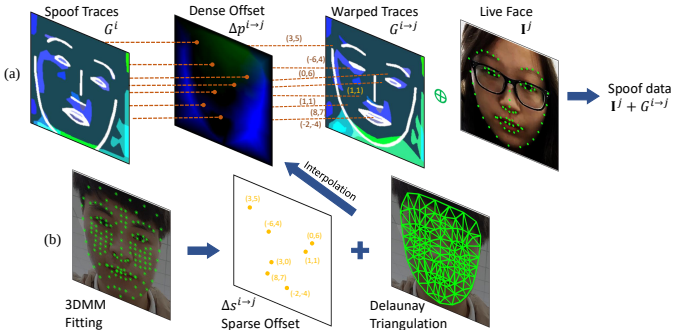


Fig. 6: The online 3D warping layer. (a) Given the corresponding dense offset, we warp the spoof trace and add them to the target live face to create a new spoof. E.g. pixel (x, y) with offset $(3, 5)$ is warped to pixel$(x + 3, y + 5)$ in the new image. (b) To obtain a dense offsets from the spare offsets of the selected face shape vertices, Delaunay triangulation interpolation is adopted.

First, the live reconstruction of a spoof face $\mathbf{I}^i$ can be expressed as:

$$G^i = G(\mathbf{I}^i)[\mathbf{p}_0], \tag{13}$$

where $\mathbf{p}^0 = \{(0, 0), (0, 1), ..., (255, 255)\} \in \mathbb{R}^{256\times256\times2}$ enumerates pixel locations in $\mathbf{I}^i$. To align the spoof traces while synthesizing a new spoof face, a dense offset $\Delta\mathbf{p}^{i\to j} \in \mathbb{R}^{256\times256\times2}$ is required to indicate the deformation between face $\mathbf{I}^i$ and face $\mathbf{I}^j$. A discrete deformation can be acquired from the distances of the corresponding facial landmarks between two faces. During the data preparation, we use [61] to fit a 3DMM model and extract the 2D locations of Q facial vertices for each face:

$$\mathbf{s} = \{(x_0, y_0), (x_1, y_1), ..., (x_N, y_N)\} \in \mathbb{R}^{Q\times2}. \tag{14}$$

A sparse offset on the corresponding vertices can then be computed as $\Delta\mathbf{s}^{i\to j} = \mathbf{s}^j - \mathbf{s}^i$. To convert the sparse offset $\Delta\mathbf{s}^{i\to j}$ to the dense offset $\Delta\mathbf{p}^{i\to j}$, we apply a triangulation interpolation:

$$\Delta\mathbf{p}^{i\to j} = \text{Tri}(\mathbf{p}^0, \mathbf{s}^i, \Delta\mathbf{s}^{i\to j}), \tag{15}$$

where $\text{Tri}(\cdot)$ is the interpolation, $\mathbf{s}^i$ denotes the vertex locations, $\Delta\mathbf{s}^{i\to j}$ are the vertex values, and we adopt Delaunay triangulation. The warping operation can be denoted as:

$$G^{-i\to j} = G^-(\mathbf{I}^j|\mathbf{I}^i)[\mathbf{p}^0 + \Delta\mathbf{p}^{i\to j}], \tag{16}$$

where the offset $\Delta\mathbf{p}^{i\to j}$ applies to all subject i related elements $\{\mathbf{T}^i_A, \mathbf{I}^i, \mathbf{P}^i\}$. Since the offset $\Delta\mathbf{p}^{i\to j}$ is typically composed of fractional numbers, we implement the bilinear interpolation to sample the fractional pixel locations. We select $Q = 140$ vertices to cover the face region so that they can represent non-rigid deformation, due to pose and expression. As the pixel values in the warped face are a linear combination of pixel values of the triangulation vertices, this entire process is differentiable. This process is illustrated in Fig. 6.

3.4 Multi-scale Discriminators

Motivated by [62], we adopt multiple discriminators at different resolutions (*e.g.*, 32, 96, and 256) in our GAN architecture. We follow the design of PatchGAN [63], which essentially is a fully convolutional network. Fully convolutional networks are shown

Algorithm 1: STDN+ Training Iteration.

Input: live faces $\mathbf{I}_{live}$ and facial landmarks $\mathbf{s}_{live}$, spoof faces $\mathbf{I}_{spoof}$ and facial landmarks $\mathbf{s}_{spoof}$, ground truth depth map $\mathbf{M}_0$, preliminary mask $\mathbf{P}_0$;
Output: reconstructed live $\hat{\mathbf{I}}_{live}$, synthesized spoof $\hat{\mathbf{I}}_{spoof}$, spoof traces $\{\mathbf{T}^l_A, \mathbf{P}^l, \mathbf{I}^l_P, \mathbf{T}^s_A, \mathbf{P}^s, \mathbf{I}^s_P\}$, depth maps $\{\mathbf{M}^l, \mathbf{M}^s\}$;

while *iteration* $<$ *max_iteration* **do**
- `// training step 1`
- **1**: compute $\mathbf{T}^l_A, \mathbf{P}^l, \mathbf{I}^l_P \leftarrow G(\mathbf{I}_{live})$ and compute $\mathbf{T}^s_A, \mathbf{P}^s, \mathbf{I}^s_P \leftarrow G(\mathbf{I}_{spoof})$;
- **2**: estimate the depth map $\mathbf{M}^l$, $\mathbf{M}^s$;
- **3**: compute losses L_{depth}, L_P, L_R;
- `// training step 2`
- **4**: compute $\hat{\mathbf{I}}_{low}$, $\hat{\mathbf{I}}_{mid}$, $\hat{\mathbf{I}}_{hi}$ from $\mathbf{T}^s_A$, $\mathbf{P}^s$, $\mathbf{I}^s_P$ and $\mathbf{I}_{spoof}$ (Eqn. 8);
- **5**: compute warping offset $\Delta\mathbf{p}^{s\to l}$ from $\mathbf{s}_{live}$, $\mathbf{s}_{spoof}$ (Eqn. 15);
- **6**: compute $\hat{\mathbf{I}}_{spoof}$ from warped $\mathbf{T}^{s\to l}_A$, $\mathbf{P}^{s\to l}$ and $\mathbf{I}_{live}$ (Eqn. 16);
- **7**: send $\mathbf{I}_{live}$, $\mathbf{I}_{spoof}$, $\hat{\mathbf{I}}_{low}$, $\hat{\mathbf{I}}_{mid}$, $\hat{\mathbf{I}}_{hi}$, $\hat{\mathbf{I}}_{spoof}$ to discriminators;
- **8**: compute the adversarial loss for generator L_G and for discriminators L_D;
- `// training step 3`
- **9**: create harder samples $\mathbf{I}_{hard}$ from $\mathbf{T}^{s\to l}_A$, $\mathbf{P}^{s\to l}$ and $\mathbf{I}_{live}$ with random perturbation on traces;
- **10**: compute $\mathbf{T}^h_A, \mathbf{P}^h, \mathbf{I}^h_P \leftarrow G(\mathbf{I}_{hard})$;
- **11**: compute depth map $\mathbf{M}^h$ for $\mathbf{I}_{hard}$;
- **12**: compute losses L_S, L_H;
- `// back propagation`
- **13**: back-propagate the losses from step $3, 8, 12$ to corresponding parts and update the network;

end

to be effective to not only synthesize high-quality images [62], [63], but also tackle face anti-spoofing problems [18]. For each discriminator, we adopt the same structure but do not share the weights.

As shown in Fig. 4, we use in total 4 discriminators in our work: D_1, working in the lowest resolution of 32, focuses on low frequency elements since the higher-frequency traces are erased by downsampling. D_2, working at the resolution of 96, focuses on the middle level content pattern. D_3 and D_4, working on the highest resolution of 256, focus on the fine texture details. Our preliminary version resizes real and synthetic samples $\{\mathbf{I}, \hat{\mathbf{I}}\}$ to different resolutions and assign to each discriminator. To remove semantic ambiguity and provide correspondence to the trace components, we instead assign the hierarchical reconstruction from Eqn. 12 to the discriminators: we send low frequency pairs $\{\mathbf{I}_{live}, \hat{\mathbf{I}}_{low}\}$ to D_1, middle frequency pairs $\{\mathbf{I}_{live}, \hat{\mathbf{I}}_{mid}\}$ to D_2, high frequency pairs $\{\mathbf{I}_{live}, \hat{\mathbf{I}}_{hi}\}$ to D_3, and real/synthetic spoof $\{\mathbf{I}_{spoof}, \hat{\mathbf{I}}_{spoof}\}$ to D_4. Each discriminator outputs a 1-channel map in the range of $[0, 1]$, where 0 denotes fake and 1 denotes real.

3.5 Loss Functions and Training Steps

We utilize multiple loss functions to supervise the learning of depth maps and spoof traces. Each training iteration consists of three training steps. The first two steps compose the GAN-based weakly-supervised trace learning. The third step composes the fully-supervised trace learning on the synthesized data. We first introduce the loss functions, followed by how they are used in the training steps.

3.5.1 Weakly-supervised Trace Learning

Trace regularization: Based on Eqn. 7 (with $\lambda = 1$), we regularize the intensity of additive traces $\mathbf{T}_A$ and the inpainting region $\mathbf{P}$. The regularizer loss is denoted as:

$$L_R = \mathbb{E}_{\mathbf{i}\sim\mathcal{L}\cup\mathcal{S}}\Big[\|\mathbf{T_A}\|^2_F + \|\mathbf{P}\|^2_F\Big]. \tag{17}$$

Inpainting mask loss: We leverage the additive traces and prior knowledge of spoof attacks to supervise inpainting masks. The inpainting mask loss consists of two terms. The first term encourages certain regions to inpaint. As inpainting step aims to relax the intensity constraint of certain spoofing regions to obtain more realistic spoof traces, additive trace regions with larger magnitude are more likely to need further inpainting process. Hence, the first term adopts a $\mathcal{L}$-2 norm between the inpainting region $\mathbf{P}$ and the region where the additive trace is larger than a threshold β.

The second term discourages certain region to inpaint based on prior knowledge. We observe some image regions that should not be inpainted. For instance, the inpainting region for funny eye glasses should not appear in the lower part of a face. The inpainting region for mask attack should not appear in the background. Shown in Fig. 7, we provide a preliminary mask $\mathbf{P}_0$ to indicate region that should not be inpainted, and adopt a normalized $\mathcal{L}$-2 norm on the masked inpainting region $\mathbf{P}\cdot\mathbf{P}_0$. The inpainting mask loss is formed as:

$$L_P = \mathbb{E}_{\mathbf{i}\sim\mathcal{S}}\Big[\|\mathbf{P}^i - (\mathbf{T}^i_A > \beta)\|^2_F + \frac{\|\mathbf{P}^i\cdot\mathbf{P}^i_0\|^2_F}{\|\mathbf{P}^i_0\|^2_F}\Big]. \tag{18}$$

Adversarial loss: We employ the LSGANs [64] on reconstructed live faces and synthesized spoof faces. It encourages the reconstructed live to look similar to real live from domain $\mathcal{L}$, and the synthesized spoof faces to look similar to faces from domain $\mathcal{S}$:

$$\begin{aligned} L_G = \mathbb{E}_{i\sim\mathcal{L},j\sim\mathcal{S}}\Big[&\|D_1(\hat{\mathbf{I}}^j_{low})-\mathbf{1}\|^2_F + \|D_2(\hat{\mathbf{I}}^j_{mid})-\mathbf{1}\|^2_F + \\ &\|D_3(\hat{\mathbf{I}}^j_{hi})-\mathbf{1}\|^2_F + \|D_4(\hat{\mathbf{I}}^{j\to i}_{spoof})-\mathbf{1}\|^2_F\Big]. \end{aligned} \tag{19}$$

The adversarial loss for discriminators encourages $D(\cdot)$ to distinguish between real live *vs.* reconstructed live, and real spoof

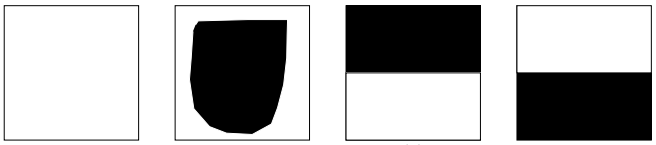


Fig. 7: Preliminary mask $\mathbf{P}_0$ for the negative term in inpainting mask loss. White pixels denote 1 and black pixels denote 0. White indicates the area should not be inpainted. $\mathbf{P}_0$ for: (a) print, replay; (b) 3D mask and makeup; (c) partial attacks that cover the eye portion; (d) partial attacks that cover the mouth portion.

vs. synthesized spoof:

$$L_D = \mathbb{E}_{i\sim\mathcal{L},j\sim\mathcal{S}}\Big[\|D_1(\mathbf{I}^i)-\mathbf{1}\|_F^2 + \|D_2(\mathbf{I}^i)-\mathbf{1}\|_F^2 + \|D_3(\mathbf{I}^i)-\mathbf{1}\|_F^2 + \|D_4(\mathbf{I}^j)-\mathbf{1}\|_F^2 + \|D_1(\hat{\mathbf{I}}^j_{low})\|_F^2 + \|D_2(\hat{\mathbf{I}}^j_{mid})\|_F^2 + \|D_3(\hat{\mathbf{I}}^j_{hi})\|_F^2 + \|D_4(\hat{\mathbf{I}}^{j\to i}_{spoof})\|_F^2\Big]. \quad (20)$$

Depth map loss: We follow the auxiliary FAS [18] to estimate an auxiliary depth map $\mathbf{M}$, where the depth ground truth $\mathbf{M}_0$ for a live face contains face-like shape and the depth for spoof should be zero. We apply the $\mathcal{L}$-1 norm on this loss as:

$$L_{depth} = \frac{1}{K^2}\mathbb{E}_{i\sim\mathcal{L}\cup\mathcal{S}}\|\mathbf{M}^i - \mathbf{M}^i_0\|_F, \quad (21)$$

where $K = 32$ is the size of $\mathbf{M}$. We apply the dense face alignment [61] to estimate the 3D shape and render the depth ground truth $\mathbf{M}_0$.

3.5.2 Fully-supervised Trace Learning

Synthesized spoof loss: Synthesized spoof data come with ground truth spoof traces. As a result, we define a pixelwise loss for the generator to disentangle the exact spoof traces that were added:

$$L_S = \mathbb{E}_{\mathbf{i}\sim\mathcal{L},\mathbf{j}\sim\mathcal{S}}\Big[\|G(\lceil G^{-j\to i}\rceil) - \lceil G^{j\to i}\rceil\|_F^1\Big], \quad (22)$$

where $G^{j\to i}$ is the overall effect of $\{\mathbf{P}^j, \mathbf{I}^j_P, \mathbf{B}^j, \mathbf{C}^j, \mathbf{T}^j\}$ after warping to subject i, and $\lceil\cdot\rceil$ is the `stop_gradient` operation. Without stopping the gradient, $G^{j\to i}$ may collapse to 0.

Depth map loss for "harder" samples: We send the "harder" synthesized spoof data to depth estimation network to improve the data diversity, and hope to increase the FAS model's generalization:

$$L_H = \frac{1}{K^2}\mathbb{E}_{i\sim\hat{\mathcal{S}}}\Big[\|\mathbf{M}^i - \mathbf{M}^i_0\|_F\Big], \quad (23)$$

where $\hat{\mathcal{S}}$ denotes the domain of synthesized spoof faces.

3.5.3 Training steps and total loss:

Each training iteration has 3 training steps. In the training step 1, live faces $\mathbf{I}_{live}$ and spoof faces $\mathbf{I}_{spoof}$ are fed into generator $G(\cdot)$ to disentangle the spoof traces. The spoof traces are used to reconstruct the live counterpart $\hat{\mathbf{I}}_{live}$ and synthesize new spoof $\hat{\mathbf{I}}_{spoof}$. The generator is updated with respect to the depth map loss L_{depth}, adversarial loss L_G, inpainting mask loss L_P, and regularizer loss L_R:

$$L = \alpha_1 L_R + \alpha_2 L_P + \alpha_3 L_G + \alpha_4 L_{depth}. \quad (24)$$

In the training step 2, the discriminators are supervised with the adversarial loss L_D to compete with the generator. In the training step 3, $\mathbf{I}_{live}$ and $\hat{\mathbf{I}}_{hard}$ are fed into the generator with the ground truth label and trace to minimize the synthesized spoof loss L_S and depth map loss L_H:

$$L = \alpha_5 L_S + \alpha_6 L_H, \quad (25)$$

where α_1-α_6 are the weights to balance the multitask training. To note that, we send the original live faces $\mathbf{I}_{live}$ with $\hat{\mathbf{I}}_{hard}$ for a balanced mini-batch, which is important when computing the moving average in the batch normalization layer. We execute all 3 steps in each minibatch iteration, but reduce the learning rate for discriminator step by half. The whole training process is depicted in Alg. 1.

Protocol	Method	APCER (%)	BPCER (%)	ACER (%)
1	STASN [21]	1.2	2.5	1.9
	Auxiliary [18]	1.6	1.6	1.6
	DeSpoof [24]	1.2	1.7	1.5
	DRL [65]	1.7	0.8	1.3
	STDN [29]	0.8	1.3	1.1
	CDCN [57]	0.4	1.7	1.0
	HMP [66]	0.0	1.6	0.8
	CDCN++ [57]	0.4	**0.0**	**0.2**
	Ours	**0.0**	0.8	0.4
2	DeSpoof [24]	4.2	4.4	4.3
	Auxiliary [18]	2.7	2.7	2.7
	DRL [65]	1.1	3.6	2.4
	STASN [21]	4.2	**0.3**	2.2
	STDN [29]	2.3	1.6	1.9
	HMP [66]	2.6	0.8	1.7
	CDCN [57]	0.4	1.7	1.5
	CDCN++ [57]	1.8	0.8	1.3
	Ours	**1.2**	1.3	**1.3**
3	DeSpoof [24]	4.0 ± 1.8	3.8 ± 1.2	3.6 ± 1.6
	Auxiliary [18]	2.7 ± 1.3	3.1 ± 1.7	2.9 ± 1.5
	STDN [29]	1.6 ± 1.6	4.0 ± 5.4	2.8 ± 3.3
	STASN [21]	4.7 ± 3.9	$\mathbf{0.9 \pm 1.2}$	2.8 ± 1.6
	HMP [66]	2.8 ± 2.4	2.3 ± 2.8	2.5 ± 1.1
	CDCN [57]	2.4 ± 1.3	2.2 ± 2.0	2.3 ± 1.4
	DRL [65]	2.8 ± 2.2	1.7 ± 2.6	2.2 ± 2.2
	CDCN++ [57]	1.7 ± 1.5	2.0 ± 1.2	$\mathbf{1.8 \pm 0.7}$
	Ours	$\mathbf{1.7 \pm 1.4}$	2.2 ± 3.5	1.9 ± 2.3
4	Auxiliary [18]	9.3 ± 5.6	10.4 ± 6.0	9.5 ± 6.0
	STASN [21]	6.7 ± 10.6	8.3 ± 8.4	7.5 ± 4.7
	CDCN [57]	4.6 ± 4.6	9.2 ± 8.0	6.9 ± 2.9
	DeSpoof [24]	5.1 ± 6.3	6.1 ± 5.1	5.6 ± 5.7
	HMP [66]	2.9 ± 4.0	7.5 ± 6.9	5.2 ± 3.7
	CDCN++ [57]	4.2 ± 3.4	5.8 ± 4.9	5.0 ± 2.9
	DRL [65]	5.4 ± 2.9	$\mathbf{3.3 \pm 6.0}$	4.8 ± 6.4
	STDN [29]	2.3 ± 3.6	5.2 ± 5.4	3.8 ± 4.2
	Ours	$\mathbf{2.3 \pm 3.6}$	4.2 ± 5.4	$\mathbf{3.6 \pm 4.2}$

TABLE 1: The evaluation on four protocols in OULU-NPU. **Bold** indicates the best score in each protocol.

3.5.4 Final Scoring

In the testing phase, we use the norm of the depth map and the intensity of spoof traces for real vs. spoof classification:

$$\text{score} = \frac{1}{2K^2}\|\mathbf{M}\|_1 + \frac{\alpha_0}{2N^2}(\|\mathbf{B}\|_1 + \|\mathbf{C}\|_1 + \|\mathbf{T}\|_1 + \|\mathbf{P}|_1), \quad (26)$$

where α_0 is the weight for the spoof trace.

4 Experiments

In this section, we first introduce the experimental setup, and then present the results in the known, unknown, and open-set spoof scenarios, with comparisons to respective baselines. Next, we quantitatively evaluate the spoof traces by performing a spoof medium classification, and conduct an ablation study on each design in the proposed method. Finally, we provide visualization results on the spoof trace disentanglement, new spoof synthesis and t-SNE visualization.

Protocol	Method	APCER (%)	BPCER (%)	ACER (%)
1	Auxiliary [18]	3.6	3.6	3.6
	STASN [21]	–	–	1.0
	Meta-FAS-DR [41]	0.5	0.5	0.5
	HMP [66]	0.6	0.2	0.5
	DRL [65]	0.1	0.5	0.3
	CDCN [57]	0.1	0.2	0.1
	CDCN++ [57]	0.1	0.2	0.1
	Ours	**0.0**	**0.0**	**0.0**
2	Auxiliary [18]	0.6 ± 0.7	0.6 ± 0.7	0.6 ± 0.7
	Meta-FAS-DR [41]	0.3 ± 0.3	0.3 ± 0.3	0.3 ± 0.3
	STASN [21]	–	–	0.3 ± 0.1
	HMP [66]	0.1 ± 0.2	0.2 ± 0.0	0.1 ± 0.1
	DRL [65]	0.1 ± 0.2	0.1 ± 0.1	0.1 ± 0.0
	CDCN [57]	0.0 ± 0.0	0.1 ± 0.1	0.1 ± 0.0
	CDCN++ [57]	0.0 ± 0.0	0.1 ± 0.1	0.0 ± 0.1
	Ours	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
3	STASN [21]	–	–	12.1 ± 1.5
	Auxiliary [18]	8.3 ± 3.8	8.3 ± 3.8	8.3 ± 3.8
	Meta-FAS-DR [41]	8.0 ± 5.0	7.4 ± 5.7	7.7 ± 5.3
	DRL [65]	9.4 ± 6.1	1.8 ± 2.6	5.6 ± 4.4
	HMP [66]	2.6 ± 0.9	2.3 ± 0.5	2.5 ± 0.7
	CDCN [57]	1.7 ± 0.1	1.8 ± 0.1	1.7 ± 0.1
	CDCN++ [57]	$\mathbf{2.0 \pm 0.3}$	$\mathbf{1.8 \pm 0.1}$	$\mathbf{1.9 \pm 0.2}$
	Ours	13.1 ± 9.4	1.6 ± 0.6	7.4 ± 4.3

TABLE 2: The evaluation on three protocols in SiW Dataset. We compare with the top 7 performances.

	Train CASIA	Test Replay	Train Replay	Test CASIA
LBP-TOP [34]	50.1		47.0	
STASN [21]	31.5		30.9	
Auxiliary [18]	27.6		28.4	
CDCN [57]	15.5		32.6	
Ours	17.1		27.7	

TABLE 3: Cross testing on CASIA-MFSD vs. Replay-Attack.

Method	CASIA			Replay			MSU-MFSD			Overall
	Video	Cut Photo	Wrap Photo	Video	Digital Photo	Print Photo	Print Photo	HR Video	Mobile Video	
DTN [19]	90.0	97.3	97.5	99.9	99.9	99.6	81.6	99.9	97.5	95.9 ± 6.2
CDCN [57]	98.5	99.9	99.8	100	99.4	99.9	70.8	100	100	96.5 ± 9.6
Ours	96.4	100	98.9	100	100	99.8	85.5	99.9	99.3	97.8 ± 4.7

TABLE 4: AUC (%) of the model testing on CASIA, Replay, and MSU-MFSD.

4.1 Experimental Setup

Databases We conduct experiments on three major databases: Oulu-NPU [67], SiW [18], and SiW-M [19]. Oulu-NPU and SiW include print/replay attacks, while SiW-M includes 13 spoof types. We follow all the existing testing protocols and compare with SOTA methods. Similar to most prior works, we only use the face region for training and testing.

Evaluation metrics Two common metrics are used in this work for comparison: EER and APCER/BPCER/ACER. EER describes the theoretical performance and predetermines the threshold for making decisions. APCER/BPCER/ACER [68] describe the practical performance given a predetermined threshold. For both evaluation metrics, lower value means better performance. The threshold for APCER/BPCER/ACER is computed from either training set or validation set. In addition, we also report the True Positive Rate (TPR) at a given False Positive Rate (FPR). This metric describes the spoof detection rate at a strict tolerance to live errors, which is widely used to evaluate real-world systems [69]. In this work, we report TPR at FPR$= 0.5\%$. For TPR, the higher the better.

Parameter setting STDN+ is implemented in Tensorflow with an initial learning rate of $5e$-5. We train in total $150,000$ iterations with a batch size of 8, and decrease the learning rate by a ratio of 10 every $45,000$ iterations. In one mini-batch, we have the same number of live and spoof samples, randomly selected. Spoof samples are equally sampled from different spoof types. If Step 3 is enabled, our network estimates the spoof traces from the spoof samples, and applies the traces to the live samples to synthesize the additional spoof samples. We initialize the weights with $[0, 0.02]$ normal distribution. $\{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\}$ are set to be $\{100, 1e\text{-}4, 1, 5, -10, 10, 1\}$, and $\beta = 0.1$. α_0 is empirically determined from the training or validation set. We use the open-source face alignment [70] and 3DMM fitting [61] to crop the face and provide 140 landmarks.

4.2 Anti-Spoofing for Known Spoof Types

Oulu-NPU Oulu-NPU [67] is a commonly used face anti-spoofing benchmark due to its high-quality data and challenging testing protocols. Tab. 1 shows our anti-spoofing performance on Oulu-NPU, compared with SOTA algorithms. Our method achieves the best overall performance on this database. Compared with our preliminary version [29], we demonstrate improvements in all 4 protocols, with significant improvement on protocol 1 and protocol 3, *i.e.*, reducing the ACER by 63.6% and 32.1% respectively. Compared with the SOTA, our approach achieves similar best performances on the first three protocols and outperforms the SOTA on the fourth protocol, which is the most challenging one. To note that, in protocol 3 and protocol 4, the performances of testing camera 6 are much lower than those of cameras 1-5: the ACER for camera 6 are 6.4% and 10.2%, while the average ACER for the other cameras are 1.0% and 2.0% respectively. Compared with other cameras, we notice that camera 6 has stronger sensor noises and our model recognizes them as unknown spoof traces, which leads to an increased false negative rate (*i.e.*, BPCER). How to separate sensor noises from spoof traces can be an important future research topic.

SiW SiW [18] is another recent high-quality database. It includes fewer capture cameras but more spoof mediums and environment variations, such as pose, illumination, and expression. The comparison on three protocols is shown in Tab. 2. We outperform the previous works on the first two protocols and rank in the middle on protocol 3. Protocol 3 aims to test the performance of unknown spoof detection, where the model is trained on one spoof attack (print or replay) and tested on the other. As we can see from Fig.8-9, the traces of print and replay are significantly different, where the replay traces are more on the high-frequency part (*i.e.*, trace component **T**) and the print traces are more on the low-frequency part (*i.e.*, trace component **S**). These pattern divergence leads to the adaption gap of our method while training on one attack and testing on the other.

SiW-M SiW-M [19] contains a large diversity of spoof types, including print, replay, 3D mask, makeup, and partial attacks. This allows us to have a comprehensive evaluation of the proposed approach with different spoof attacks. To use SiW-M for known spoof detection, we randomly split the data of all types into train/test set with a ratio of 60% vs. 40%, and the results are shown in Tab. 5. Compared to the preliminary version [29], our method outperforms on most spoof types as well as the overall EER performance by 47.9% relatively, which demonstrates the superiority of our anti-spoofing on known spoof attacks.

For experiments on SiW-M (protocol I, II, and III), we additionally report the TPR at FNR equal to 0.5%. While EER and ACER provide the theoretical evaluation, the users in real-world applications care more about the true spoof detection rate under a

Metrics(%)	Method	Replay	Print	3D Mask					Makeup			Partial Attacks			Overall
				Half	Silic.	Trans.	Paper	Mann.	Ob.	Im.	Cos.	Funny.	Papergls.	Paper	
ACER	Auxiliary [18]	5.1	5.0	5.0	10.2	5.0	9.8	6.3	19.6	5.0	26.5	5.5	5.2	5.0	6.3
	SDTN [29]	3.2	3.1	3.0	9.0	3.0	3.4	4.7	**3.0**	3.0	24.5	4.1	3.7	3.0	4.1
	Step1	6.1	5.4	5.4	5.4	5.4	5.4	5.4	22.7	5.4	26.8	5.4	5.5	5.4	10.9
	Step1+Step2 w/ single trace	8.7	7.8	7.8	7.8	7.8	7.8	7.8	25.0	7.9	28.8	7.8	7.8	7.8	13.8
	Step1+Step2	4.1	3.9	3.9	3.9	4.0	3.9	4.0	13.5	4.0	25.1	3.9	3.9	3.9	4.6
	Multiplicative model	3.0	2.7	2.5	2.5	2.1	5.8	2.9	18.1	1.2	29.2	2.3	1.7	2.2	3.6
	Step1+Step2+Step3 (Ours)	**3.2**	**1.4**	**1.0**	**2.3**	**1.3**	**2.9**	**2.5**	12.4	**1.2**	**18.5**	**1.7**	**0.4**	**1.6**	**2.8**
EER	Auxiliary [18]	4.7	0.0	1.6	10.5	4.6	10.0	6.4	12.7	0.0	19.6	9.3	7.5	0.0	6.7
	SDTN [29]	2.1	2.2	0.0	7.2	0.1	3.9	4.8	**0.0**	0.0	19.6	5.3	5.4	**0.0**	4.8
	Step1	3.8	2.7	1.5	2.7	1.9	**1.8**	2.4	15.1	0.7	28.7	4.1	4.9	1.0	4.3
	Step1+Step2 w/ single trace	6.7	5.3	0.8	**1.5**	1.4	3.3	3.2	21.5	1.0	27.1	6.5	6.1	1.5	5.8
	Multiplicative model	2.5	2.8	0.0	2.5	1.5	5.0	2.4	10.6	0.0	30.4	2.2	1.8	0.0	3.5
	Step1+Step2	2.4	3.1	0.4	2.6	1.2	3.0	2.4	9.5	0.4	23.5	1.1	0.5	0.6	2.8
	Step1+Step2+Step3 (Ours)	**2.5**	**1.0**	**0.0**	2.1	**1.0**	1.9	**2.2**	8.2	**0.0**	**18.5**	**0.8**	**0.0**	0.4	**2.5**
TPR@ FPR=.5%	SDTN [29]	**90.1**	76.1	80.7	71.5	62.3	74.4	85.0	**100**	100	33.8	49.6	30.6	97.7	70.4
	Step1	43.8	43.3	47.2	44.5	62.9	54.8	55.4	16.7	90.6	31.5	60.3	56.7	77.1	59.3
	Step1+Step2 w/ single trace	58.9	76.8	97.6	**94.2**	94.9	66.3	78.3	13.3	94.1	49.1	62.4	58.5	92.1	74.8
	Multiplicative model	82.0	85.1	99.3	70.3	95.1	71.2	89.4	38.3	100	47.9	77.9	94.7	**100**	84.7
	Step1+Step2	84.7	74.7	100	70.1	**96.6**	77.5	89.6	36.9	100	40.1	96.3	99.4	99.4	89.7
	Step1+Step2+Step3 (Ours)	85.7	**85.4**	**100**	76.6	96.3	**80.2**	**93.8**	41.1	**100**	**55.8**	**98.1**	**100**	99.8	**91.2**

TABLE 5: The evaluation and ablation study on SiW-M Protocol I: known spoof detection.

Metrics (%)	Method	Replay	Print	3D Mask					Makeup			Partial Attacks			Average
				Half	Silic.	Trans.	Paper	Mann.	Ob.	Im.	Cos.	Fun.	Papergls.	Paper	
APCER	Auxiliary [18]	23.7	7.3	27.7	18.2	97.8	8.3	16.2	100.0	18.0	16.3	91.8	72.2	0.4	38.3 ± 37.4
	LBP+SVM [67]	19.1	15.4	40.8	20.3	70.3	0.0	4.6	96.9	35.3	**11.3**	53.3	58.5	0.6	32.8 ± 29.8
	DTL [19]	**1.0**	0.0	0.7	24.5	58.6	0.5	3.8	73.2	13.2	12.4	17.0	17.0	0.2	17.1 ± 23.3
	CDCN [57]	8.2	6.9	8.3	7.4	20.5	5.9	5.0	43.5	1.6	14.0	24.5	18.3	1.2	12.7 ± 11.7
	SDTN [29]	1.6	**0.0**	**0.5**	**7.2**	9.7	0.5	**0.0**	96.1	0.0	21.8	**14.4**	**6.5**	0.0	12.2 ± 26.1
	CDCN++ [57]	9.2	6.0	4.2	7.4	18.2	**0.0**	5.0	39.1	0.0	14.0	23.3	14.3	0.0	10.8 ± 11.2
	HMP [66]	12.4	5.2	8.3	9.7	13.6	**0.0**	2.5	**30.4**	0.0	12.0	22.6	15.9	1.2	$\mathbf{10.3 \pm 9.1}$
	Ours	10.0	4.9	5.3	16.7	**3.5**	2.0	2.8	92.8	**0.0**	37.5	33.7	23.2	0.2	17.9 ± 25.8
BPCER	LBP+SVM [67]	22.1	21.5	21.9	21.4	20.7	23.1	22.9	21.7	12.5	22.2	18.4	20.0	22.9	21.0 ± 2.9
	DTL [19]	18.6	11.9	29.3	12.8	13.4	8.5	23.0	11.5	9.6	16.0	21.5	22.6	16.8	16.6 ± 6.2
	SDTN [29]	14.0	14.6	13.6	18.6	18.1	8.1	13.4	10.3	9.2	17.2	27.0	35.5	11.2	16.2 ± 7.6
	CDCN [57]	9.3	8.5	13.9	10.9	21.0	**3.1**	7.0	45.0	2.3	16.2	26.4	20.9	5.4	14.6 ± 11.7
	CDCN++ [57]	12.4	8.5	14.0	13.2	19.4	7.0	6.0	45.0	1.6	14.0	24.8	20.9	3.9	14.6 ± 11.4
	HMP [66]	13.2	6.2	13.1	10.8	16.3	3.9	**2.3**	34.1	**1.6**	13.9	23.2	17.1	2.3	12.2 ± 9.4
	Auxiliary [18]	10.1	6.5	10.9	11.6	**6.2**	7.8	9.3	11.6	9.3	7.1	**6.2**	8.8	10.3	8.9 ± 2.0
	Ours	**3.8**	**6.3**	**4.4**	**5.5**	11.3	3.5	6.0	**6.6**	1.8	**2.7**	6.5	**8.0**	**1.1**	$\mathbf{5.7 \pm 2.8}$
ACER	LBP+SVM [67]	20.6	18.4	31.3	21.4	45.5	11.6	13.8	59.3	23.9	16.7	35.9	39.2	11.7	26.9 ± 14.5
	Auxiliary [18]	16.8	6.9	19.3	14.9	52.1	8.0	12.8	55.8	13.7	**11.7**	49.0	40.5	5.3	23.6 ± 18.5
	DTL [19]	9.8	6.0	15.0	18.7	36.0	4.5	13.4	48.1	11.4	14.2	19.3	19.8	8.5	16.8 ± 11.1
	CDCN [57]	8.7	7.7	11.1	**9.1**	20.7	4.5	5.9	44.2	2.0	15.1	25.4	19.6	3.3	13.6 ± 11.7
	SDTN [29]	7.8	7.3	7.1	12.9	13.9	4.3	6.7	53.2	4.6	19.5	20.7	21.0	5.6	14.2 ± 13.2
	CDCN++ [57]	10.8	7.3	9.1	10.3	18.8	3.5	5.6	42.1	0.8	14.0	24.0	17.6	1.9	12.7 ± 11.2
	HMP [66]	12.8	5.7	10.7	10.3	14.9	**1.9**	**2.4**	**32.3**	**0.8**	**12.9**	22.9	16.5	1.7	$\mathbf{11.2 \pm 9.2}$
	Ours	**6.9**	**5.6**	**4.8**	11.1	**7.4**	2.7	4.4	49.7	0.9	20.1	**20.1**	**15.6**	**0.6**	11.5 ± 13.2
EER	LBP+SVM [67]	20.8	18.6	36.3	21.4	37.2	7.5	14.1	51.2	19.8	16.1	34.4	33.0	7.9	24.5 ± 12.9
	Auxiliary [18]	14.0	4.3	11.6	12.4	24.6	7.8	10.0	72.3	10.1	**9.4**	21.4	18.6	4.0	17.0 ± 17.7
	DTL [19]	10.0	**2.1**	14.4	18.6	26.5	5.7	9.6	50.2	10.1	13.2	19.8	20.5	8.8	16.1 ± 12.2
	CDCN [57]	8.2	7.8	8.3	**7.4**	20.5	5.9	5.0	47.8	1.6	14.0	24.5	18.3	1.1	13.1 ± 12.6
	SDTN [29]	7.6	3.8	8.4	13.8	14.5	5.3	4.4	35.4	0.0	19.3	21.0	20.8	1.6	12.0 ± 10.0
	CDCN++ [57]	9.2	5.6	**4.2**	11.1	19.3	5.9	5.0	43.5	0.0	14.0	23.3	14.3	**0.0**	11.9 ± 11.8
	HMP [66]	13.4	5.2	8.3	9.7	13.6	5.8	**2.5**	**33.8**	0.0	14.0	23.3	16.6	1.2	11.3 ± 9.5
	Ours	**5.2**	4.4	4.4	10.1	**8.6**	**2.6**	4.3	47.2	**0.0**	19.6	**18.6**	**12.4**	0.7	$\mathbf{10.6 \pm 12.6}$
TPR@ FPR=.5%	SDTN [29]	45.0	40.5	45.7	36.7	11.7	40.9	74.0	0.0	67.5	16.0	13.4	9.4	62.8	35.7 ± 23.9
	Ours	**55.1**	**46.4**	**57.3**	**65.1**	**33.0**	**91.7**	**76.7**	**0.0**	**100.0**	**46.4**	**31.8**	**15.4**	**97.7**	$\mathbf{53.7 \pm 31.8}$

TABLE 6: The evaluation on SiW-M Protocol II: unknown spoof detection.

Label \ Predict	Live	Print1	Print2	Replay1	Replay2
Live	56(−4)	1(+1)	1(+1)	1(+1)	1(+1)
Print1	0	43(+2)	11(+9)	3(−8)	3(−3)
Print2	0	9(−25)	48(+37)	1(−8)	2(−4)
Replay1	1(−9)	2(−1)	3(+3)	51(+38)	3(−28)
Replay2	1(−7)	2(−5)	2(+2)	3(−3)	52(+13)

TABLE 7: Confusion matrices of spoof mediums classification based on spoof traces. The results are compared with the previous method [24]. Green represents improvement over [24]. Red represents performance drop.

Label \ Predict	Live	Print	Replay	Masks	Makeup	Partial
Live	116	6	6	3	0	0
Print	1	40	1	3	0	1
Replay	3	1	32	1	0	1
Masks	3	1	1	90	0	3
Makeup	3	0	0	0	36	0
Partial	2	0	0	2	0	146

TABLE 8: Confusion matrices of 6-class spoof traces classification on SiW-M database.

given live detection error rate, and hence TPR can better reflect how well the model can detect one or a few spoof attacks in practices. As shown in Tab. 5, we improve the overall TDR of our preliminary version [29] by 29.5%.

When we evaluate the model on different testing protocols, data plays a big role on the performance difference. SiW Protocol 3 above only contains 90 subjects with data collected in a controlled environment. When a model is trained from scratch, there is not enough knowledge that can lead to a good generalization. In

contrast, SiW-M contains over 600 subjects with data collected in a wide range of environment variations. Hence, it's more likely to learn better feature representation for the zero-shot scenario. For the future study, we believe warmstarting on large pretrained databases such as ImageNet and CelebA-Spoof might help to improve the generalization.

4.3 Anti-Spoofing for Unknown and Open-set Spoofs

Another important aspect is to test the anti-spoofing performance on unknown spoof. To use SiW-M for unknown spoof detection, The work [19] defines the leave-one-out testing protocols, termed as SiW-M Protocol II. In this protocol, each model (*i.e.*, one column in Tab. 6) is trained with 12 types of spoof attacks (as known attacks) plus the 80% of the live faces, and tested on the remaining 1 attack (as unknown attack) plus the 20% of live faces. As shown in Tab. 6, our STDN+ achieves significant improvement over our preliminary version, with relatively 11.7% on the overall EER, 19.0% on the overall ACER, 50.4% on the overall TPR. Specifically, we reduce the EERs of half mask, paper glasses, transparent mask, replay attack, and partial paper relatively by 47.6%, 40.4%, 37.7%, 31.6%, 56.3%, respectively. Overall, compared with the top 7 performances, we outperform the SOTA performance of EER/TPR and achieve comparable ACER. Among all, the detection of silicone mask, paper-crafted mask, mannequin head, impersonation makeup, and partial paper attacks are relatively good, with the detection accuracy (*i.e.*, TPR@FNR=0.5%) above 65%. Obfuscation makeup is the most challenging one with TPR of 0%, where we predict all the spoof samples as live. This is due to the fact that the makeup looks very similar to the live faces, while being dissimilar to any other spoof types. However, once we obtain a few samples, our model can quickly recognize the spoof traces on the eyebrow and cheek, synthesize new spoof samples, and successfully detect the attack (TPR=41.1% in Tab. 5).

Moreover, in the real-world scenario, the testing samples can be either a known spoof attack or an unknown one. Thus, we propose SiW-M Protocol III to evaluate this open-set testing situation. In Protocol III, we first follow the train/test split from protocol I, and then further remove one spoof type as the unknown attack. During the testing, we test on the entire unknown spoof samples as well the test split set of the know spoof samples. The results are reported in Tab. 9. Compared to the SOTA face anti-spoofing method [18], our approach substantially outperforms it in all three metrics. In addition, we implement a cross-testing on CASIA and Replay, and a cross-type testing on CASIA, Replay, and MSU-MFSD. The results are shown in Tab. 3, and Tab. 4. As shown in the tables, our model achieves the top performance and two experiments can serve as a complement to the open set protocol.

4.4 Spoof Traces Classification

To quantitatively evaluate the spoof trace disentanglement, we perform a spoof medium classification on the disentangled spoof traces and report the classification accuracy. The spoof traces should contain spoof medium-specific information, so that they can be used for clustering without seeing the face. To make a fair comparison with [24], we remove the additional spoof type information from the preliminary mask $\mathbf{P}_0$. That is, for this specific experiment, we only use the additive traces $\{\mathbf{B}, \mathbf{C}, \mathbf{T}\}$ to learn the trace classification. After $\{\mathbf{B}, \mathbf{C}, \mathbf{T}\}$ finish training with only binary labels, we fix STDN+ and apply a simple CNN (*i.e.*, AlexNet) on the estimated additive traces to do a supervised spoof medium classification. We follow the same 5-class testing protocol in [24] in Oulu-NPU Protocol 1. We report the classification accuracy as the ratio between correctly predicted samples from all classes and all testing samples. Shown in Tab. 7. Our model can achieve a 5-class classification accuracy of 83.3%. If we treat two print attacks as the same class and two replay as the same class, our model can achieve a 3-class classification accuracy of 92.0%. Compared with the prior method [24], we show an improvement of 29% on the 5-class model. In addition, we train the same CNN on the original images instead of the estimated spoof traces for the same spoof medium classification task, and the classification accuracy can only reach 80.6%. This further demonstrates that the estimated traces do contain significant information to distinguish different spoof mediums.

We also execute the spoof traces classification task on more spoof types in SiW-M database. We leverage the train/test split on SiW-M Protocol 1. We first train the STDN+ till convergence, and use the estimated traces from the training set to train the trace classification network. We explore the 6-class scenario, shown in Tab. 8. Our 6-class model can achieve the classification accuracy of 92.0%. Since the traces are more distinct among different spoof types, this performance is even better than 5-class classification on print/replay scenario in Oulu-NPU Protocol 1. In contrast, 6-class classification on the original image can only reach 76.4%. This further demonstrates that STDN+ can estimate spoof traces that contain significant information of spoof mediums and can be applied to multiple spoof types.

4.5 Ablation Study

In this section, we show the importance of each design of our proposed approach on the SiW-M Protocol I, in Tab.5. Our baseline is the auxiliary FAS [18], without the temporal module. It consists of the backbone encoder and depth estimation network. When including the image decomposition, the baseline becomes the training step 1 in Alg. 1, as the traces are not activated without the training step 2. To validate the effectiveness of GAN training, we report the results from the baseline model with our GAN design, denoted as Step1+Step2. We also provide the control experiment where the traces are represented by a single component to demonstrate the effectiveness of the proposed 5-element trace representation. This model is denoted as Step1+Step2 with single trace. In addition, we evaluate the effect of training with more synthesized data via enabling the training step 3 as Step1+Step2+Step3, which is our final approach.

As shown in Tab. 5, the baseline model (Auxiliary) can achieve a decent performance of EER 6.7%. Adding image decomposition to the baseline (Step 1) can improve the EER from 6.7% to 4.3%, but more live samples are predicted with higher scores, causing a worse ACER. Adding simple GAN design (Step1+Step2 with single trace) may lead to a similar EER performance of 5.8%, but based on the TPR (59.3% $\rightarrow$ 74.8%) its practical performance may be improved. With the proper physics-guided trace disentanglement, we can improve the EER to 2.8% and TPR to 89.7%. And our final design can achieve the performance of HTER 2.8%, EER 2.5%, and TPR 91.2%. Compared with our preliminary version, the EER is improved by 47.9%, HTER is improved by 31.7% and TPR is improved by 29.5%.

Comparison to other modeling multiplicative model is a commonly used model in 3D face reconstruction, object illumination

Metrics (%)	Method	Replay	Print	3D Mask					Makeup			Partial Attacks			Overall
				Half	Silic.	Trans.	Paper	Mann.	Ob.	Im.	Cos.	Funny.	Papergls.	Paper	
ACER	Auxiliary [18]	6.7	5.6	8.5	7.5	11.6	6.7	6.4	8.9	5.7	6.1	14.3	15.9	5.4	8.4 ± 3.4
	Ours	**4.7**	**3.5**	**3.4**	**3.3**	**6.4**	**2.6**	**3.8**	**7.0**	**2.3**	**3.2**	**10.7**	**7.3**	**3.2**	$\mathbf{4.7 \pm 2.4}$
EER	Auxiliary [18]	6.4	5.6	7.7	6.5	10.3	6.1	6.1	8.4	5.1	6.3	15.3	13.1	5.7	7.9 ± 3.2
	Ours	**4.1**	**2.8**	**3.4**	**3.1**	**5.6**	**3.6**	**3.0**	**6.7**	**2.2**	**3.4**	**10.2**	**8.6**	**2.2**	$\mathbf{4.5 \pm 2.5}$
TPR@ FPR=.5%	Auxiliary [18]	60.4	65.5	64.4	70.4	47.5	67.0	71.6	64.3	75.1	69.8	45.8	47.8	62.9	62.5 ± 9.7
	Ours	**87.4**	**78.7**	**81.0**	**84.5**	**69.0**	**86.3**	**84.7**	**85.0**	**91.0**	**89.3**	**66.6**	**64.4**	**91.1**	$\mathbf{81.6 \pm 9.2}$

TABLE 9: The evaluation on SiW-M Protocol III: openset spoof detection.

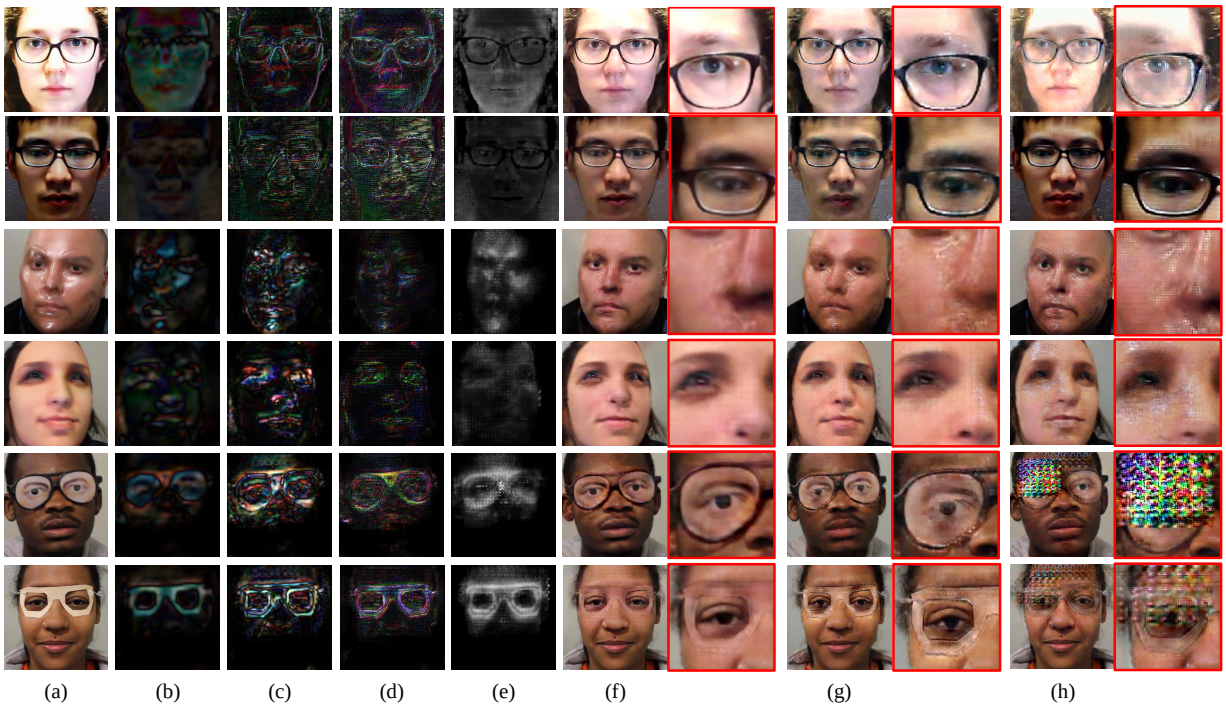


Fig. 8: Examples of each spoof trace components. (a) the input sample faces. (b) **B**. (c) **C**. (d) **T**. (e) **P**. (f) the final live counterpart reconstruction and zoom-in details. (g) results from [29]. (h) results from Step1+Step2 with a single trace representation.

and etc. We attempt to brute-forcely formulate the spoof trace modeling via a multiplicative model:

$$\hat{\mathbf{I}}_{live} = \mathbf{W} \cdot \mathbf{I}_{spoof} + \mathbf{T}_{\mathrm{A}}, \tag{27}$$

where W is the multiplicative spoof component and T_A is the additive component. For simplicity, we don't bring the 3D geometry factors (e.g., surface normal, illumination modeling) into the equation, but we believe the above model is enough to validate the effectiveness of the multiplicative model. To synthesize, we can revert the above equation as:

$$\hat{\mathbf{I}}^{i \to j}_{spoof} = (\mathbf{I}^{j}_{live} - \mathbf{T}^{i}_{\mathrm{A}})/\mathbf{W}^{i}. \tag{28}$$

where i, j are two subjects. We train a network with this multiplicative model and keep other settings the same.

The quantitative results on SiW-M Protocol I are shown in the Tab. 5. As we can see from the table, the overall performance is higher than the baseline generative model (i.e., Step1 + Step2 w/ single trace) but lower than our proposed modeling (i.e. Step1 + Step2 / Step1 + Step2 + Step3). Some spoof trace visualizations from this modeling are shown in Fig. 13. We can tell that both the proposed model and multiplicative model have a decent performance on the live reconstruction. However, the new spoof synthesis from multiplicative model has a poor quality, and it potentially lead to a worse performance. Theoretically, there is no physical meaning attached to the component $\mathbf{W}$, which makes the multiplicative model only suitable for live reconstruction but not new spoof synthesis. In comparison, the propose additive and inpainting model can well handle both the live reconstruction and spoof synthesis, and it achieve better performances.

Relation between spoof trace and score : To validate the relation of spoof trace intensity and the spoofness score, we execute a 3-round reverse spoof trace disentanglement on SiW-M Protocol 1. The input of the first round is the original testing images, the input of the second round is the original images with 50% of the estimated spoof traces (from the first round) removed, and the third round is the original images with 100% estimated spoof traces removed. The 1st-round average spoof score for live faces is 0.89, the second round is 0.93, and the third round is 0.97. The 1st-round average spoof score for spoof faces is 0.11, the second round is 0.47, and the third round is 0.97. Based on the results, we can tell that the spoof trace intensity is positively correlated with the spoofness score. Therefore, if a sample contains weaker spoof traces, it becomes a harder sample for the model to detect it as spoof.

4.6 Visualization

Spoof trace components In Fig.8, we provide illustration of each spoof trace component. Strong color distortion (low-frequency

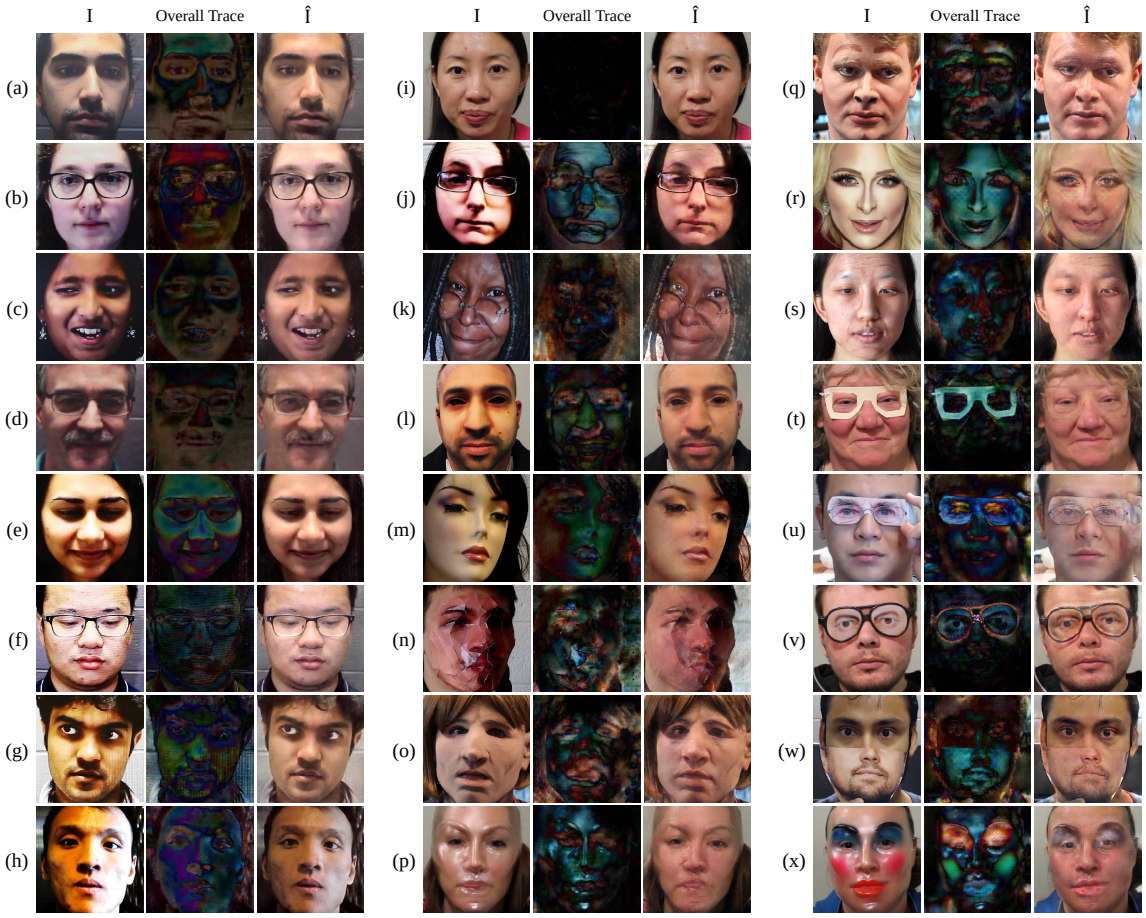


Fig. 9: Examples of spoof trace disentanglement on SiW (a-h) and SiW-M (i-x). (a)-(d) items are print attacks and (e)-(h) items are replay attacks. (i)-(x) items are live, print, replay, half mask, silicone mask, paper mask, transparent mask, obfuscation makeup, impersonation makeup, cosmetic makeup, paper glasses, partial paper, funny eye glasses, and mannequin head. The first column is the input face, the second column is the overall spoof trace ($\mathbf{I} - \hat{\mathbf{I}}$), the third column is the reconstructed live.

trace) shows up in the print attacks. Moiré patterns in the replay attack are well detected in the high-frequency trace. The local specular highlights in transparent mask are well presented in the low- and mid-frequency components, and the inpainting process further fine-tunes the most highlighted area. For the two glasses attacks, the color discrepancy is corrected in the low-frequency trace, and the sharp edges are corrected in the mid- and high-frequency traces. Each component shows a consistent semantic meaning on different spoof samples, and this successful trace disentanglement can lead to better final visual results. As shown on the right side of Fig. 8, we compare with our preliminary version [29] and the ablated GAN design with a single trace representation. The result of single trace representation shows strong artifacts on most of the live reconstruction. The multi-scale from our preliminary version has already shown a large visual quality improvement, but still have some spoof traces (*e.g.*, glass edges) remained in the live reconstruction. In contrast, our approach can further handle the missing traces and achieve better visualization.

Live reconstruction In Fig. 9, we show more examples from different spoof types in SiW and SiW-M databases. The overall trace is the exact difference between the input face and its live reconstruction. For the live faces, the trace is zero, and for the spoof faces, our method removes spoof traces without unnecessary changes, such as identity shift, and make them look like live faces. For example, strong color distortion shows up in print/replay attacks (Fig. 9a-h) and some 3D mask attacks (Fig. 9l-o). For makeup attacks (Fig. 9q-s), the fake eyebrows, lipstick, artificial wax, and cheek shade are clearly detected. The folds and edges (Fig. 9t-w) are well detected and removed in paper-crafted masks, paper glasses, and partial paper attacks.

Spoof synthesis Additionally, we show examples of new spoof synthesis using the disentangled spoof traces, which is an important contribution of this work. As shown in Fig. 10, the spoof traces can be precisely transferred to a new face without changing the identity of the target face. Due to the additional inpainting process, spoof attacks such as transparent mask and partial attacks can be better attached to the new live face. Thanks to the proposed 3D warping layer, the geometric discrepancy between the source spoof trace and the target face can be corrected during the synthesis. Especially on the second source spoof, the right part of the traces is successfully transferred to the new live face while the left side remains to be still live. It demonstrates that our trace regularization can suppress unnecessary artifacts generated by the network. Both the live reconstruction results in Fig. 9 and the spoof synthesis results in Fig. 10 demonstrate that our approach disentangles visually convincing spoof traces that help face anti-spoofing.

Spoof trace removing process As shown in Fig. 12, we illustrate the effects of trace components by progressively removing them one by one. For the replay attack, the spoof sample comes with

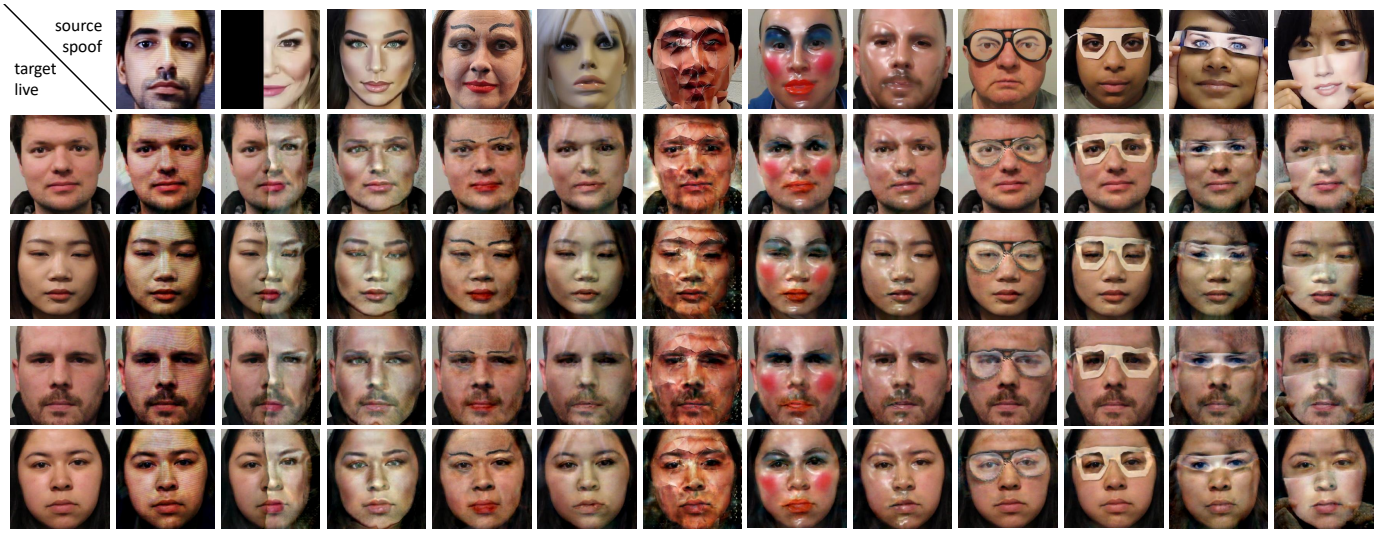


Fig. 10: Examples of the spoof data synthesis. The first row are the source spoof faces, the first column are the target live faces, and the remaining are the synthesized spoof faces from the live face with the corresponding spoof traces.

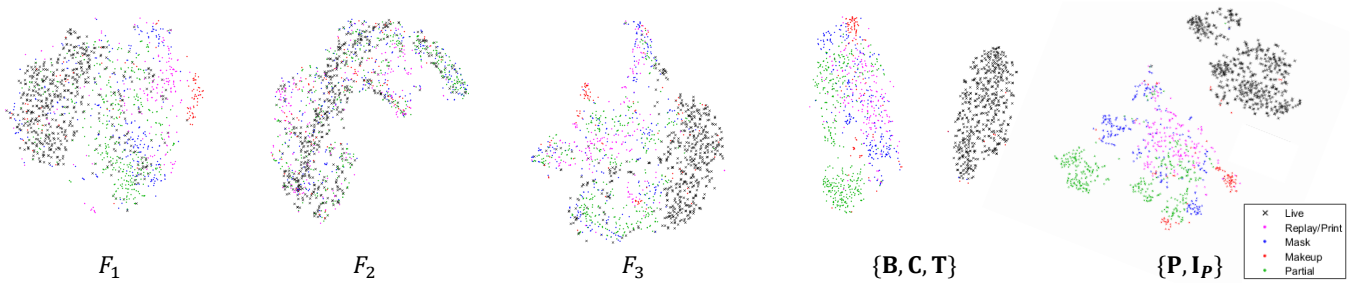


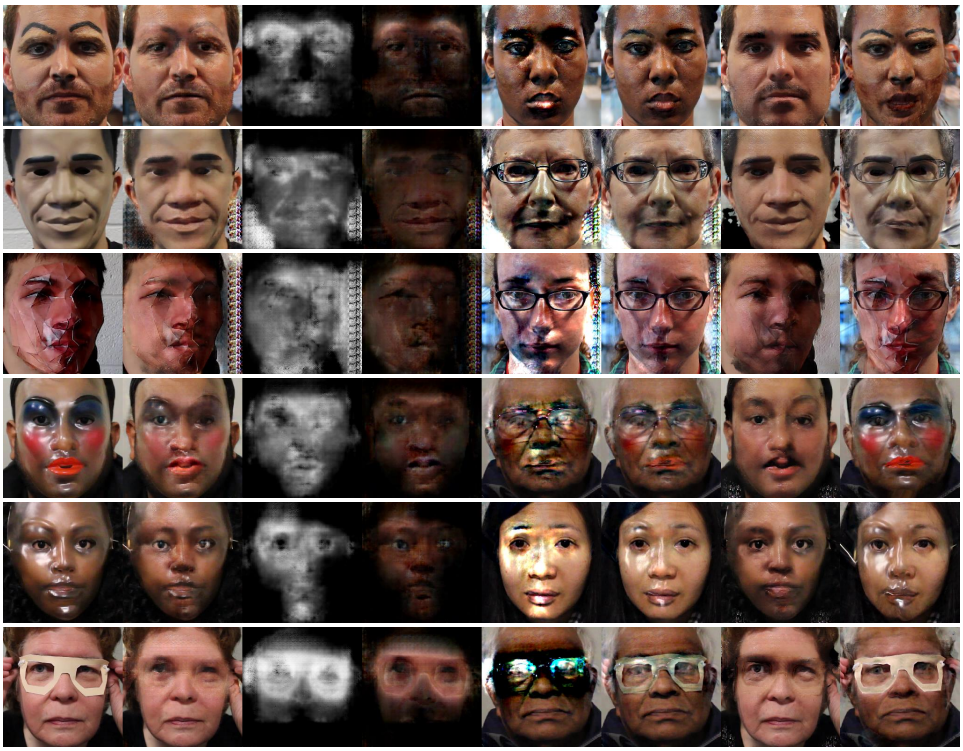
Fig. 11: The tSNE visualization of features from different scales and layers. The first 3 visualization are from the encoder feature F_1,F_2,F_3, and the last 2 visualization are from the features that produce $\{\mathbf{B}, \mathbf{C}, \mathbf{T}\}$ and $\{\mathbf{P}, \mathbf{I}_P\}$.

Fig. 12: The illustration of removing the disentangled spoof trace components one by one. The estimated spoof trace elements of input spoof (the first column) are progressively removed in the order of $\mathbf{B}, \mathbf{C}, \mathbf{T}, \mathbf{T_P}$. The last column shows the reconstructed live image after removing all three additive trace components and the inpainting trace. (a) Replay attack; (b) Makeup attack; (c) Mask attack; (d) Paper glasses attack.

strong over-exposure as well as clear Moiré pattern. Removing the low-frequency trace can effectively correct the over-exposure and color distortion caused by the digital screen. And removing the texture pattern in the high-frequency trace can peel off the high-frequency grid effect and reconstruct the live counterpart.

For the makeup attack, since there is no strong color range bias, removing estimated low-frequency trace would mainly remove the lip-stick color and fake eyebrow, but in the meantime bring a few artifacts at the edges. Next, while removing the content pattern, the shadow on the cheek and the fake eyebrows are adequately lightened. Finally, removing the texture pattern would significantly correct the spoof traces from artificial wax, eyeliner, and shadow on the cheek. Similarly, in mask and partial attacks, the reconstruction will be gradually refined as we removing components one by one.

t-SNE visualization We use t-SNE [71] to visualize the encoder features $\mathbf{F}_1$,$\mathbf{F}_2$,$\mathbf{F}_3$, and the features that produce $\{\mathbf{B}, \mathbf{C}, \mathbf{T}\}$ and $\{\mathbf{P}, \mathbf{I}_P\}$. The t-SNE is able to project the output of features from different scales and layers to 2D by preserving the KL divergence distance. As shown in Fig. 11, among the three feature scales in the encoder, F_1 and F_3 have a similar separability, while F_2 has the worst. It shows that the most useful features are coming from either the high-frequency texture (*i.e.*, F_1) or the low-frequency texture (*i.e.*, F_3) The features for additive traces $\{\mathbf{B}, \mathbf{C}, \mathbf{T}\}$ are well-clustered as semantic sub-groups of live, makeup, mask, and partial attacks. As we know the inpainting masks for live samples are close to zero, the feature for inpainting traces $\{\mathbf{P}, \mathbf{I}_P\}$ shows the inpainting process mostly update the partial attacks, and then some makeup attacks and mask attacks, *i.e.*, the green dots being further away from the black dots means they have greater magnitude. This validates our prior knowledge of the inpainting process.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 13: The comparison of different trace modeling. (a) Input (b) Live reconstruction from multiplicative model (c) $\mathbf{W}$ (d) $\mathbf{T}_A$ (e) Synthesis based on reverse multiplicative model (f) Synthesis based on addition ($\mathbf{I}_{\text{spoof}} - \hat{\mathbf{I}}_{\text{live}}$) (g) Live reconstruction from the proposed model (h) Synthesis from our proposed model.

5 Conclusions

This work proposes a new spoof traces disentanglement network (STDN+) to tackle the challenging problem of disentangling spoof traces from the input faces. We model the spoof trace disentanglement as a combined process of additive step and inpainting step. With the spoof traces, we reconstruct the live faces as well as synthesize new spoofs. To correct the geometric discrepancy in synthesis, we propose a 3D warping layer to deform the traces. The disentanglement not only improves the SOTA of face anti-spoofing in known, unknown, and open-set spoof settings, but also provides visual evidence to support the model's decision.

Acknowledgments

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2017-17020200004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

[1] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister, "Video face replacement," in *TOG*. ACM, 2011.

[2] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," *arXiv preprint arXiv:1905.08233*, 2019.

[3] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt, "State of the art on monocular 3D face reconstruction, tracking, and applications," in *Computer Graphics Forum*. Wiley Online Library, 2018.

[4] D. Deb, J. Zhang, and A. K. Jain, "Advfaces: Adversarial face synthesis," *arXiv preprint arXiv:1908.05008*, 2019.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[7] J. F. Boylan, "Will deep-fake technology destroy democracy?" in *The New York Times*, 2018.

[8] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of RGB videos," in *CVPR*. IEEE, 2016.

[9] J. Bigun, H. Fronthaler, and K. Kollreider, "Assuring liveness in biometric identity authentication by real-time face tracking," in *International Conference on Computational Intelligence for Homeland Security and Personal Safety (CIHSPS)*. IEEE, 2004.

[10] R. W. Frischholz and A. Werner, "Avoiding replay-attacks in a face recognition system using head-pose estimation," in *International SOI Conference. (Cat. No. 03CH37443)*. IEEE, 2003.

[11] S. A. Schuckers, "Spoofing and anti-spoofing measures," *Information Security technical report*, 2002.

[12] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analysis," in *ICIP*. IEEE, 2015, pp. 2636–2640.

[13] ——, "Face antispoofing using speeded-up robust features and fisher vector encoding," *Signal Processing Letters*, 2016.

[14] K. Patel, H. Han, and A. K. Jain, "Secure face unlock: Spoof detection on smartphones," *TIFS*, 2016.

[15] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "LBP-TOP based countermeasure against face spoofing attacks," in *ACCV*. Springer, 2012.

[16] J. Komulainen, A. Hadid, and M. Pietikäinen, "Context based face anti-spoofing," in *BTAS*. IEEE, 2013, pp. 1–8.

[17] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based CNNs," in *IJCB*. IEEE, 2017.

[18] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *CVPR*. IEEE, 2018.

[19] Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu, "Deep tree learning for zero-shot face anti-spoofing," in *CVPR*. IEEE, 2019.

[20] R. Shao, X. Lan, J. Li, and P. C. Yuen, "Multi-adversarial discriminative deep domain generalization for face presentation attack detection," in *CVPR*. IEEE, 2019.

[21] X. Yang, W. Luo, L. Bao, Y. Gao, D. Gong, S. Zheng, Z. Li, and W. Liu, "Face anti-spoofing: Model matters, so does data," in *CVPR*. IEEE, 2019.

[22] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *arXiv preprint arXiv:1408.5601*, 2014.

[23] Z. Xu, S. Li, and W. Deng, "Learning temporal features using lstm-cnn architecture for face anti-spoofing," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 2015.

[24] A. Jourabloo, Y. Liu, and X. Liu, "Face de-spoofing: Anti-spoofing via noise modeling," in *ECCV*. Springer, 2018.

[25] J. Stehouwer, A. Jourabloo, Y. Liu, and X. Liu, "Noise modeling, synthesis and classification for generic object anti-spoofing," in *CVPR*, 2020.

[26] H. Feng, Z. Hong, H. Yue, Y. Chen, K. Wang, J. Han, J. Liu, and E. Ding, "Learning generalized spoof cues for face anti-spoofing," *arXiv preprint arXiv:2005.03922*, 2020.

[27] "Explainable Artificial Intelligence (XAI)," https://www.darpa.mil/program/explainable-artificial-intelligence.

[28] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, 2020.

[29] Y. Liu, J. Stehouwer, and X. Liu, "On disentangling spoof traces for generic face anti-spoofing," in *ECCV*, 2020.

[30] Y. Liu and C. Shu, "A comparison of image inpainting techniques," in *Sixth International Conference on Graphic and Image Processing (ICGIP 2014)*, 2015.

[31] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000.

[32] K. Kollreider, H. Fronthaler, M. I. Faraj, and J. Bigun, "Real-time face detection and motion analysis with application in "liveness" assessment," *TIFS*, 2007.

[33] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *ICCV*. IEEE, 2007.

[34] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?" in *ICB*. IEEE, 2013, pp. 1–8.

[35] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *IJCB*. IEEE, 2011.

[36] J. Yang, Z. Lei, S. Liao, and S. Z. Li, "Face liveness detection with component dependent descriptor," in *ICB*. IEEE, 2013.

[37] L. Feng, L.-M. Po, Y. Li, X. Xu, F. Yuan, T. C.-H. Cheung, and K.-W. Cheung, "Integration of image quality and motion cues for face anti-spoofing: A neural network approach," *Journal of Visual Communication and Image Representation*, 2016.

[38] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2016.

[39] K. Patel, H. Han, and A. K. Jain, "Cross-database face antispoofing with robust feature representation," in *CCBR*. Springer, 2016.

[40] Y. Qin, C. Zhao, X. Zhu, Z. Wang, Z. Yu, T. Fu, F. Zhou, J. Shi, and Z. Lei, "Learning meta model for zero-and few-shot face anti-spoofing," *arXiv preprint arXiv:1904.12490*, 2019.

[41] C. Zhao, Y. Qin, Z. Wang, T. Fu, and H. Shi, "Meta anti-spoofing: Learning to learn in face anti-spoofing," *arXiv preprint arXiv:1904.12490*, 2019.

[42] R. Shao, X. Lan, and P. C. Yuen, "Regularized fine-grained meta face anti-spoofing," *arXiv preprint arXiv:1911.10771*, 2019.

[43] H.-P. Huang, D. Sun, Y. Liu, W.-S. Chu, T. Xiao, J. Yuan, H. Adam, and M.-H. Yang, "Adaptive transformers for robust few-shot cross-domain face anti-spoofing," *arXiv preprint arXiv:2203.12175*, 2022.

[44] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning GAN for pose-invariant face recognition," in *CVPR*. IEEE, 2017.

[45] Z. Zhang, L. Tran, X. Yin, Y. Atoum, J. Wan, N. Wang, and X. Liu, "Gait recognition via disentangled representation learning," in *CVPR*. IEEE, 2019.

[46] F. Liu, D. Zeng, Q. Zhao, and X. Liu, "Disentangling features in 3D face shapes for joint face reconstruction and recognition," in *CVPR*. IEEE, 2018.

[47] L. Tran and X. Liu, "On learning 3d face morphable model from in-the-wild images," *T-PAMI*, Jan 2021.

[48] P. Esser, E. Sutter, and B. Ommer, "A variational U-Net for conditional appearance and shape generation," in *CVPR*. IEEE, 2018.

[49] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *CVPR*, 2018.

[50] T. H. Thai, R. Cogranne, and F. Retraint, "Camera model identification based on the heteroscedastic noise model," *TIP*, 2013.

[51] T. H. Thai, F. Retraint, and R. Cogranne, "Camera model identification based on the generalized noise model in natural images," *Digital Signal Processing*, 2016.

[52] Y. Wang, L. Bondi, P. Bestagini, S. Tubaro, D. J. Edward Delp *et al.*, "A counter-forensic method for cnn-based camera model identification," in *CVPRW*, 2017.

[53] C. Chen, Z. Xiong, X. Liu, and F. Wu, "Camera trace erasing," in *CVPR*, 2020.

[54] Y. Wu, W. AbdAlmageed, and P. Natarajan, "Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9543–9552.

[55] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in *CVPR*, 2020.

[56] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015.

[57] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao, "Searching central difference convolutional networks for face anti-spoofing," in *CVPR*. IEEE, 2020.

[58] Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu, "Presentation attack detection for face in mobile phones," *Selfie Biometrics*, 2019.

[59] J. Guo, X. Zhu, J. Xiao, Z. Lei, G. Wan, and S. Z. Li, "Improving face anti-spoofing by 3D virtual synthesis," *arXiv preprint arXiv:1901.00488*, 2019.

[60] H. Chang, J. Lu, F. Yu, and A. Finkelstein, "Paired cycleGAN: Asymmetric style transfer for applying and removing makeup," in *CVPR*. IEEE, 2018.

[61] Y. Liu, A. Jourabloo, W. Ren, and X. Liu, "Dense face alignment," in *ICCV Workshops*. IEEE, 2017.

[62] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *CVPR*. IEEE, 2018.

[63] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*. IEEE, 2017.

[64] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *ICCV*. IEEE, 2017.

[65] K.-Y. Zhang, T. Yao, J. Zhang, Y. Tai, S. Ding, J. Li, F. Huang, H. Song, and L. Ma, "Face anti-spoofing via disentangled representation learning," in *ECCV*, 2020.

[66] Z. Yu, X. Li, X. Niu, J. Shi, and G. Zhao, "Face anti-spoofing with human material perception," in *ECCV*, 2020.

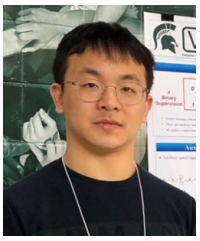
[67] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, "OULU-NPU: A mobile face presentation attack database with real-world variations." IEEE, 2017.

[68] ISO/IEC JTC 1/SC 37 Biometrics, "Information technology biometric presentation attack detection part 1: Framework. international organization for standardization," https://www.iso.org/obp/ui/iso, 2016.

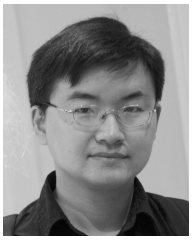
[69] "IARPA research program Odin)," https://www.iarpa.gov/index.php/research-programs/odin.

[70] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3d facial landmarks)," in *ICCV*. IEEE, 2017.

[71] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, 2008.



Yaojie Liu is a research scientist at Google Research. He received the Ph.D. degree in Computer Science and Engineering from Michigan State University in 2021. He received the M.S. in Computer Science from the Ohio State University in 2016 and the B.S. in Communication Engineering from University of Electronic Science and Technology of China in 2014. His research areas of interest are security of face biometric systems (e.g., face anti-spoofing, digital manipulation attack, adversarial attack), 3D face modeling, face representation & analysis, XAI, image systhesis, and multi-model modeling.



Xiaoming Liu is a MSU Foundation Professor at the Department of Computer Science and Engineering of Michigan State University. He received the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University in 2004. Before joining MSU in Fall 2012, he was a research scientist at General Electric (GE) Global Research. His research interests include computer vision, machine learning, and biometrics. As a co-author, he is a recipient of Best Industry Related Paper Award runner-up at ICPR 2014, Best Student Paper Award at WACV 2012 and 2014, Best Poster Award at BMVC 2015, and Michigan State University College of Engineering Withrow Endowed Distinguished Scholar Award. He has been Area Chairs for numerous conferences, including CVPR, ICCV, ECCV, ICLR, NeurIPS, the Program Co-Chair of BTAS'18, WACV'18, and AVSS'21 conferences, and General Co-Chair of FG'23 conference. He is an Associate Editor of Pattern Recognition Letters, Pattern Recognition, and IEEE Transaction on Image Processing. He has authored more than 150 scientific publications, and has filed 27 U.S. patents. He is a fellow of IAPR.