Learning Meta Model for Zero- and Few-shot Face Anti-spoofing

Yunxiao Qin,[1,2] Chenxu Zhao,[2*] Xiangyu Zhu,[3] Zezheng Wang,[2] Zitong Yu,[4]
Tianyu Fu,[5] Feng Zhou,[2] Jingping Shi,[1] Zhen Lei[3]
[1]Northwestern Polytechnical University, Xian, China, [2]AIBEE, Beijing, China
[3]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, Beijing, China
[4]CMVS, University of Oulu, Oulu, Finland, [5]Winsense Technology Ltd, Beijing, China
qyxqyx@mail.nwpu.edu.cn, {cxzhao; zezhengwang; fzhou}@aibee.com, {xiangyu.zhu; zlei}@nlpr.ia.ac.cn
zitong.yu@oulu.fi, futianyu514@gmail.com, shijingping@nwpu.edu.cn

Abstract

Face anti-spoofing is crucial to the security of face recognition systems. Most previous methods formulate face anti-spoofing as a supervised learning problem to detect various predefined presentation attacks, which need large scale training data to cover as many attacks as possible. However, the trained model is easy to overfit several common attacks and is still vulnerable to unseen attacks. To overcome this challenge, the detector should: 1) learn discriminative features that can generalize to unseen spoofing types from predefined presentation attacks; 2) quickly adapt to new spoofing types by learning from both the predefined attacks and a few examples of the new spoofing types. Therefore, we define face anti-spoofing as a zero- and few-shot learning problem. In this paper, we propose a novel **A**daptive **I**nner-update **M**eta **F**ace **A**nti-**S**poofing (AIM-FAS) method to tackle this problem through meta-learning. Specifically, AIM-FAS trains a meta-learner focusing on the task of detecting unseen spoofing types by learning from predefined living and spoofing faces and a few examples of new attacks. To assess the proposed approach, we propose several benchmarks for zero- and few-shot FAS. Experiments show its superior performances on the presented benchmarks to existing methods in existing zero-shot FAS protocols.

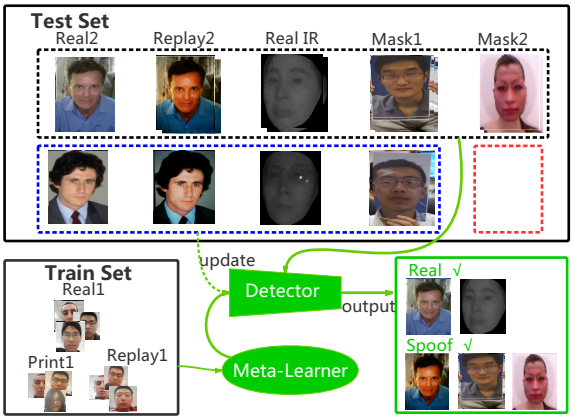

Figure 1: A zero- and few-shot FAS example. The train set contains several predefined living and spoofing types. The test set contains several faces of new emerged living and spoofing types. Zero-shot FAS is training the detector only on the train set and testing it on the test set. Whereas, few-shot FAS utilizes both the train set and a few collected faces (the blue box) for updating the detector.

1 Introduction

Face recognition is a ubiquitous technology used in industrial applications and commercial products. However, face recognition system is easy to be fooled by presentation attacks (PAs), such as printed face (print attack), face replay on digital device (replay attack), face covered by a mask (3D-mask attack), etc. As a result, face anti-spoofing (FAS) system, which detects whether the presented face is live or not, becomes essential to keep the recognition system safe.

Until now, researchers have proposed lots of hand-crafted feature based (Boulkenafet, Komulainen, and Hadid 2016; Gan et al. 2017; Lucena et al. 2017) and deep-learning based methods(Lucena et al. 2017; Xu, Li, and Deng 2015; Shao, Lan, and Yuen 2017) to discriminate spoof faces from living faces. Most of them train the detector to learn how to discriminate living and spoofing faces with numerous predefined living and spoofing faces in a supervised way.

The detectors are satisfactory at detecting the predefined PAs due to their data-driven training manner. However, when deployed in real scenarios, the FAS systems will encounter the following practical challenges.

- A variety of application scenarios and unpredictable novel PAs keep evolving. Data-driven models may give unpredictable results when faced with out-of-distribution living examples captured in new application scenarios and spoofing examples with new PAs.
- When we adapt the anti-spoofing model to new attacks, existing methods need to collect sufficient samples for training. However, it is expensive to collect labeled data for every new attack since the spoofing keeps evolving.

To overcome these challenges, we propose that FAS should be treated as an open set zero- and few-shot learn-

*Corresponding Author.

ing problem. As shown in Fig.1, *Zero-shot learning* aims to learn general discriminative features, which are effective to detect un-predicted new PAs, from the predefined PAs. *Few-shot learning* aims to quickly adapt the anti-spoofing model to new attacks by learning from both the predefined PAs and the collected very few examples of the new attack.

Zero-shot FAS problem has been studied in (Liu et al. 2019; Arashloo, Kittler, and Christmas 2017) neglecting the few-shot scene. As aforementioned, the FAS detector should solve both zero- and few-shot FAS problems. To this end, inspired by the model-agnostic meta-learning (MAML) (Finn, Abbeel, and Levine 2017), we propose a novel meta-learning based FAS method: Adaptive Inner-update Meta Face Anti-spoofing (AIM-FAS).

AIM-FAS solves the zero- and few-shot FAS problem by **F**usion **T**raining (FT) a meta-learner on zero- and few-shot FAS tasks, with **A**daptive **I**nner-**U**pdate (AIU) learning rate strategy. FT means the meta-learner is forced to focus on simultaneously learning: 1) general discriminative features to detect unseen PAs from predefined PAs, if no instance of the new PA has been collected; 2) better discriminative features to adapt to new PA from both the predefined PAs and the few instances of the new PA, once a few instances of the new PA are collected. AIU means the meta-learner inner-updates with a learn-able regular inner-update step size.

To evaluate the zero- and few-shot FAS, we propose three benchmarks to assess the FAS model's learning capability of detecting new PAs from the same domain, different domains, and different modals.

The main contributions of this paper are:

- To the best of our knowledge, we are the first to formulate FAS as a zero- and few-shot learning problem.
- To solve zero- and few-shot FAS problem, we propose a novel meta-learning based approach: Adaptive Inner-update Meta Face Anti-spoofing (AIM-FAS), which Fusion Trains (FT) a meta-learner on zero- and few-shot FAS tasks with a novel developed Adaptive Inner-Update (AIU) strategy.
- We propose three novel zero- and few-shot FAS benchmarks to validate the efficacy of AIM-FAS.
- Comprehensive experiments are conducted to show that AIM-FAS achieves state-of-the-art results on zero- and few-shot anti-spoofing benchmarks.

2 Background

2.1 Face Anti-Spoofing

Traditional FAS methods (de Freitas Pereira et al. 2012; 2013; M*ää*ttä, Hadid, and Pietik*ä*inen 2011; Patel, Han, and Jain 2016b; Boulkenafet, Komulainen, and Hadid 2017; Komulainen, Hadid, and Pietikainen 2013) usually extract hand-crafted features from the facial images and train a binary classifier to detect spoofing faces. Recently, deep learning based FAS methods (Lucena et al. 2017; Nagpal and Dubey 2018; Li et al. 2016; Patel, Han, and Jain 2016a) attract more attention. These methods commonly train a deep network to learn static discrimination between living and spoofing faces, with binary classification or depth regression supervision. Recent researches show that the depth regression supervised methods (Atoum et al. 2017; Liu, Jourabloo, and Liu 2018) outperform the binary classification methods, mainly because they provide the network with more detail information to study the spoofing cues. However, either traditional or deep learning based approaches are still sensitive to various conditions, such as illumination, blur pattern, capture camera, and presentation attack instruments. Slight change of these conditions would significantly affect the performance of the FAS detector.

2.2 Few-shot and Zero-shot Learning

Few-shot learning (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017), which aims at learning from very few instances, has attracted lots of attention. Metric learning based methods are popular to solve few-shot learning problem. These methods train a non-linear mapping function projecting images to an embedding space, and classify the image with nearest neighbor or linear classifier. Recently, meta-learning (Bengio et al. 1992; Finn, Abbeel, and Levine 2017; Nichol, Achiam, and Schulman 2018; Duan et al. 2016; Mishra et al. 2018; Grant et al. 2018; Qin et al. 2018) based methods solve few-shot learning by training a meta-learner on few-shot learning tasks. Given a few examples of new object categories, these methods train the meta-learner to recognize the new categories by memorizing (Mishra et al. 2018; Duan et al. 2016) the few examples of the new categories or updating its weight(Finn, Abbeel, and Levine 2017; Nichol, Achiam, and Schulman 2018; Qin et al. 2018).

Few-shot learning task is usually referred to N-way K-shot learning task, which contains N unseen categories for the model to recognize. Compared to conventional classification problem, each way in the task has a relatively smaller number (K) of labeled examples provided for training. In a nutshell, an N-way K-shot task provides a support set of NK labeled examples for the model to learn. In evaluation, a query set that contains several other examples from the N unseen categories is used to test the model.

Zero-shot learning which aims at to recognize unseen category with only description or semantic attributes of the new category. Similar to metric learning based few-shot learning, traditional zero-shot learning methods train a model to learn a visual-semantic embedding (Lampert, Nickisch, and Harmeling 2009; 2014; Norouzi et al. 2013). Once the embedding is trained, the instance of unseen classes can be classified in two steps. Firstly, the instance is projected into the semantic space. Secondly, it is labeled to the class which has the most similar semantic attributes.

For *Zero-shot learning task*, the model is required to recognize unseen categories by learning only from the description or semantic information of these unseen categories. In other words, the support set of the zero-shot learning task contains only the description or semantic information of these unseen categories. In this paper, we prefer to solve both zero- and few-shot FAS problems simultaneously.

3 Methodology

In this section, we detail the proposed Adaptive Inner-update Meta Face Anti-spoofing (AIM-FAS) method.

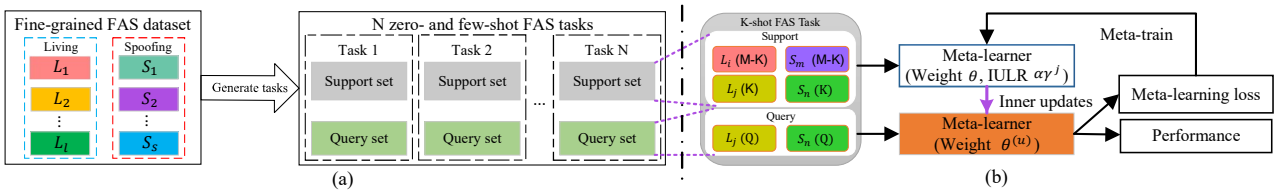

Figure 2: (a) The fine-grained FAS dataset contains several living ($L_1, L_2, ..., L_l$) and spoofing ($S_1, S_2, ..., S_s$) categories, and generates N zero- and few-shot FAS tasks. (b) The meta-learner inner-updates itself on the support set for u steps (the pink arrow), and updates its weight θ to $\theta^{(u)}$. Then we get the meta-learner's zero- and few-shot learning performance and meta-learning loss by testing the updated meta-learner on the query set. Finally, we optimize the meta-learner with the meta-learning loss. The L_j (Q) in query set of the K-shot FAS task means the query set contains Q faces from the L_j living face category.

3.1 Zero- and Few-shot FAS Problem

Zero- and few-shot FAS task We propose that there exist general discriminative features among predefined PAs and unpredicted new PAs. In other words, the knowledge in predefined living and spoofing faces can be transferred to detect new living (*e.g.* the living faces recorded in new application scenarios) and new spoofing types. Therefore, we define zero- and few-shot FAS task differently from the traditional zero- and few-shot learning task. In zero-shot FAS, the model learns the feature to recognize new living and spoofing categories from predefined living and spoofing categories. The support set in zero-shot FAS task only contains predefined living and spoofing faces. In few-shot FAS task, the model learns the feature to detect new spoofing types not only from the predefined types but also from a few examples of new living and spoofing types The support set in few-shot FAS task contains faces of not only new living and spoofing types but also predefined types.

Task generation To generate zero- and few-shot FAS tasks, we split the living and spoofing faces into fine-grained pattern, and show the fine-grained dataset structure in Fig.2(a). We show an example of K-shot FAS task in Fig.2(b), and generate the K-shot ($K >= 0$) FAS tasks in the following way: 1) sample one fine-grained living category L_i and one spoofing category S_m, from the train set. 2) sample $M - K$ faces from each of L_i and S_m. 3) re-sample one fine-grained living category L_j and one spoofing category S_n. Note that, for training tasks, L_j and S_n are sampled from the train set, and for testing tasks, they are sampled from the test set. 4) sample $K + Q$ faces from each of L_j and S_n. 5) build the query set with $2Q$ faces from L_j and S_n, and build the support set with the other $2 * (M - K) + 2 * K = 2M$ faces. In other words, L_i and S_m can be seen as the predefined categories, and L_j and S_n can be seen as the new emerged categories. In this way, we generate both zero-shot and few-shot learning tasks. When $K = 0$ (zero-shot FAS), the meta-learner learns from L_i and S_m, and predict faces from L_j and S_n. When $K > 0$ (few-shot FAS), the meta-learner learns from L_i, L_j, S_m and S_n, and predict faces from L_j and S_n.

3.2 AIM-FAS

To tackle the zero- and few-shot FAS problem, we develop our Adaptive Inner-update Meta Face Anti-Spoofing (AIM-FAS) with training a meta-learner on zero- and few-shot FAS training tasks. Furthermore, an Adaptive Inner-Update (AIU) strategy is presented to improve the performance further, as the meta-learner will inner-update more accurately on the support set with AIU. Specifically, on a given zero- or few-shot FAS task, one training iteration of the meta-learner consists of two stages.

Inner-update stage The meta-learner with weight θ inner-updates itself on the support set for several steps which can be formulated as:

$$\mathcal{L}_{s(\tau_i)}(\theta_i^{(j)}) \leftarrow \frac{1}{\|s(\tau_i)\|} \sum\nolimits_{x,y \in s(\tau_i)} l(f_{\theta_i^{(j)}}(x), y), \quad (1)$$

$$\theta_i^{(j+1)} \leftarrow \theta_i^{(j)} - \alpha \cdot \gamma^j \cdot \nabla_{\theta_i^{(j)}} \mathcal{L}_{s(\tau_i)}(\theta_i^{(j)}), \quad (2)$$

where τ_i is a randomly selected zero- or few-shot FAS training task, and $\theta_i^{(j)}$ is the meta-learner's weight after j inner-update steps. Note that, for each task τ_i, $\theta_i^{(0)} = \theta$ when $j = 0$. x and y is a pair of instance and label sampled from the support set of τ_i. $\|s(\tau_i)\|$ is the number of instances of the support set. If not otherwise specified, $\|s(\tau_i)\| = 2M$. $f_{\theta_i^{(j)}}(x)$ is the meta-learner's prediction on instance x, and $\mathcal{L}_{s(\tau_i)}(\theta_i^{(j)})$ is the meta-learner's loss on the support set.

Scalar parameter α and γ in Eq.2 are the keys to achieve AIU. Both of them are trainable. The product of α and γ^j is the inner-update learning rate (IULR). j is the meta-learner's inner-update step. The IULR changes along with the updates of j. For example, when the meta-learner inner-updates itself on the support set for the first step (j=0), the IULR is α itself. When the meta-learner inner-updates itself on the support set for the second step (j=1), the IULR turns to $\alpha \cdot \gamma^1$. With trainable α and γ, the meta-learner inner-updates with an adaptive step size. After u inner-update steps, the meta-learner update its weight from θ to $\theta_i^{(u)}$ on the support set with Eq.1 and Eq.2.

Optimizing stage The meta-learner is evaluated and optimized on the query set, which contains faces of unseen living and spoofing categories. The optimization can be formulated as:

$$\mathcal{L}_{q(\tau_i)}(\theta_i^{(u)}) \leftarrow \frac{1}{\|q(\tau_i)\|} \sum\nolimits_{x,y \in q(\tau_i)} l(f_{\theta_i^{(u)}}(x), y) \quad (3)$$

$$(\theta, \alpha, \gamma) \leftarrow (\theta, \alpha, \gamma) - \beta \cdot \nabla_{(\theta,\alpha,\gamma)} \mathcal{L}_{q(\tau_i)}(\theta_i^{(u)}) \quad (4)$$

Algorithm 1 AIM-FAS in training stage

input: K-shot ($K >= 0$) FAS training tasks Ψ_t, learning rate β, number of inner-update steps u, initial value of AIU parameters α and γ.
output: Meta-learner's weight θ, AIU parameters α and γ.
1 : initialize θ and AIU parameters α and γ.
2 : pre-train the meta-learner on the train set.
3 : while not done **do**
4 : sample batch tasks $\tau_i \in \Psi_t$
5 : **for** each of τ_i **do**
6 : $\theta_i^{(0)} = \theta$
7 : **for** $j < u$ **do**
8 : $\mathcal{L}_{s(\tau_i)}(\theta_i^{(j)}) \leftarrow \frac{1}{\|s(\tau_i)\|} \sum_{x,y \in s(\tau_i)} l(f_{\theta_i^{(j)}}(x), y)$
9 : $\theta_i^{(j+1)} \leftarrow \theta_i^{(j)} - \alpha \cdot \gamma^j \cdot \nabla_{\theta_i^{(j)}} \mathcal{L}_{s(\tau_i)}(\theta_i^{(j)})$
10: $\mathcal{L}_{q(\tau_i)}(\theta_i^{(j+1)}) \leftarrow \frac{1}{\|q(\tau_i)\|} \sum_{x,y \in q(\tau_i)} l(f_{\theta_i^{(j+1)}}(x), y)$
11: $j = j + 1$
12: **end**
13: **end**
14: $(\theta, \alpha, \gamma) \leftarrow (\theta, \alpha, \gamma)$ - $\beta \cdot \nabla_{(\theta,\alpha,\gamma)} \sum_{\tau_i} \mathcal{L}_{q(\tau_i)}(\theta_i^{(u)})$
15: end

where x and y is a pair of instance and label from the query set of task τ_i. $\|q(\tau_i)\|$ is the number of instances of the query set. If not otherwise specified, $\|q(\tau_i)\|$ is $2Q$. Note that when the meta-learner is evaluated on the query set, its weight is $\theta_i^{(u)}$, which is updated from θ with Eq.2 for u inner-update steps. Further more, in Eq.4, $\nabla_{(\theta,\alpha,\gamma)} \mathcal{L}_{q(\tau_i)}(\theta_i^{(u)})$ uses the meta-learner's loss on query to compute the gradient of θ, α and γ, but not $\theta_i^{(u)}$. β is the learning rate in the optimizing stage. By constantly training the meta-learner on lots of these zero- and few-shot learning tasks, the meta-learner is forced to learn easy fine-tuning weight θ and propriety α and γ. With weight θ and the adaptive IULR $\alpha \cdot \gamma^j$, the meta-learner updates itself accurately on the support set, and learn the discriminative features to detect unseen spoofing types.

The training process of AIM-FAS is shown in Algorithm 1 and Fig.2(b). We firstly pre-train the meta-learner to learn the prior knowledge about FAS on the train set (line 2 in Algorithm 1), and secondly meta-train the meta-learner on zero- and few-shot FAS training tasks. The testing process of AIM-FAS is shown in Algorithm 2, in which $P_{q(\tau_i)}$ is the meta-learner performance on the query set of τ_i. $X_{q(\tau_i)}$ and $Y_{q(\tau_i)}$ are the faces and labels in the query set of task τ_i.

Difference between AIM-FAS with the other traditional FAS methods The difference between AIM-FAS with the other traditional FAS is that AIM-FAS trains the meta-learner to focus on learning the discrimination for detecting new spoofing category, from the support set where contains predefined living and spoofing faces and a few or none data of the new living and spoofing categories, while traditional FAS methods train a detector to learn the discrimination for detecting predefined spoofing faces.

Fusion Train (FT) Traditionally, meta-learning methods train meta-learners independently for different K-shot ($K >$ 0) learning problems. For example, to solve 1-shot learning problem, they usually train a meta-learner on 1-shot training tasks, and to solve 5-shot learning problem, they train another meta-learner on 5-shot training tasks. In contrast, our goal is training one meta-learner to solve both zero- and few-shot FAS tasks. So, in AIM-FAS, we train the meta-learner in a Fusion Training (FT) manner, which means the meta-learner is simultaneously trained on different K-shot ($K >= 0$) FAS tasks. Specifically, the meta-learner is trained on tasks of both zero- and few-shot FAS tasks, *ie.* 0-shot, 1-shot, 2-shot, *etc.*. In our experiment, we show that FT improves AIM-FAS on both zero- and few-shot FAS tasks.

Algorithm 2 AIM-FAS in testing stage

input: K-shot FAS testing tasks Ψ_v, number of inner-update steps u, Meta-learner's weight θ, AIU parameters α and γ.
output: Meta-learner's performance *P*.
1 : for each of $\tau_i \in \Psi_v$ **do**
2 : $\theta_i^{(0)} = \theta$
3 : **for** $j < u$ **do**
4 : $\mathcal{L}_{s(\tau_i)}(\theta_i^{(j)}) \leftarrow \frac{1}{\|s(\tau_i)\|} \sum_{x,y \in s(\tau_i)} l(f_{\theta_i^{(j)}}(x), y)$
5 : $\theta_i^{(j+1)} \leftarrow \theta_i^{(j)} - \alpha \cdot \gamma^j \cdot \nabla_{\theta_i^{(j)}} \mathcal{L}_{s(\tau_i)}(\theta_i^{(j)})$
6 : $j = j + 1$
7 : **end**
8 : $P_{q(\tau_i)} \leftarrow p(f_{\theta_i^{(u)}}(X_{q(\tau_i)}), Y_{q(\tau_i)})$
9 : end
10: $P \leftarrow \frac{1}{\|\Psi_v\|} \sum_{\tau_i \in \Psi_v} P_{q(\tau_i)}$

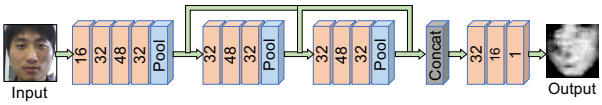

Figure 3: Network structure of AIM-FAS. The pink cube is the convolution layer, on which the number means the number of channels of its filter.

Network Depth-supervised FAS methods (Liu, Jourabloo, and Liu 2018) take advantage of the discrimination between spoofing and living faces based on 3D shape, and provide more detailed information for the FAS model to capture spoofing cues. Motivated by this, AIM-FAS trains the meta-learner to solve depth regression based zero- and few-shot FAS tasks. We build a depth regression network for AIM-FAS and name it as FAS-DR. The structure and detail of FAS-DR is shown in Fig.3. There are three cascaded blocks in the network backbone, and all their features are concatenated for predicting the facial depth. We formulate the facial depth prediction process as $\widetilde{D} = f_\theta(x)$, where $x \in \mathbb{R}^{256\times256\times3}$ is the RGB facial image, and $\widetilde{D} \in \mathbb{R}^{32\times32\times1}$ is the predicted facial depth, and θ is the network's weights.

Contrastive Depth Loss (CDL) (Wang et al. 2018) is utilized to help the meta-leaner to predict vivid facial depth. The CDL is

$$L^{contrast} = \sum_i \|k_i^{contrast} \cdot \widetilde{D} - k_i^{contrast} \cdot D\|_2^2, \quad (5)$$

where D is the generated "ground truth" facial depth label. $k_i^{contrast}$ is the kernel of CDL, and i $\in \{0,1,2,3,4,5,6,7\}$.

4 Zero- and Few-shot FAS Benchmarks

To verify AIM-FAS, we propose three **Z**ero- and **F**ew-shot FAS benchmarks:*OULU-ZF*, *Cross-ZF* and *SURF-ZF*. All three benchmarks will be released soon.

OULU-ZF is a single domain zero- and few-shot FAS benchmark and is build based on OULU-NPU. In OULU-NPU, there are 6 image capture devices, 3 kinds of living faces (living faces captured within 3 different sessions), 2 kinds of print attacks, and 2 kinds of replay attacks. All living and spoofing faces are captured with 55 people. We reorganize OULU-NPU into *OULU-ZF* and show the structure of *OULU-ZF* in Tab.1. There is no overlap between the train (seen categories) and test (unseen categories) set. The train set contains 2 kinds of living face (living 2 and 3) and 2 kinds of spoofing faces (print 1 and replay 1). Whereas, the test set contains the living 1, the print 2 and the replay 2 categories.

Cross-ZF is a cross domain zero- and few-shot FAS benchmark which is more challenging than *OULU-ZF*. It contains more varied living and spoofing categories. We build *Cross-ZF* based on several public FAS dataset. Tab.1 shows the structure of *Cross-ZF*. The train set contains 7 kinds of living faces, 4 kinds of printed faces, and 7 kinds of replayed faces, from three public dataset: CASIA-MFSD, MSU-MFSD, and SiW. The test set contains living and spoofing faces from the other three dataset: 3DMAD, Oulu-NPU, and Replay-Attack. There is no overlap between the train set and test set, and the test set contains 3D Mask faces, which are different greatly with printed and replayed faces.

SURF-ZF is a cross modal zero- and few-shot FAS benchmark. We build *SURF-ZF* based on the CASIA-SURF dataset. Structure of *SURF-ZF* is shown in Tab.3. We extract several samples from CASIA-SURF, and split these examples into train, validation, and test set. The train set contains RGB and Depth modalities, and the test/validation set contains IR and depth modalities. Each set contains all PSAIs (Living 1;Print1;Cut1-5). Based on *SURF-ZF*, we can test the model's ability of learning fast from new modalities.

5 Experiments

5.1 Experiment Setup

Performance Metrics In our experiments, AIM-FAS is evaluated by: 1) Attack Presentation Classification Error Rate ($APCER$); 2) Bona Fide Presentation Classification Error Rate ($BPCER$); 3) $ACER$ (international organization for standardization 2016), which evaluates the mean of $APCER$ and $BPCER$. 4) Area Under Curve (AUC).

Evaluation Process On all benchmarks, we evaluate the meta-learner's zero- and few-shot FAS performance in the following way: 1) train the meta-learner on the training tasks generated on the train set; 2) test the meta-learner on zero- and few-shot FAS testing tasks on the test set; 3) calculate the meta-learner's performance with Eq.6.

$$\begin{aligned} ACER_{avg} &= \sum ACER_{i=1}^{T}/T, \\ ACER &= ACER_{avg} \pm 1.96 * \sigma/\sqrt{T} \end{aligned} \tag{6}$$

Table 1: Zero- and few-shot FAS benchmark: *OULU-ZF*.

Set	Device	Subjects	PSAI
Train	Phone 1,2,4,5,6	1-20	Living 2,3; Print 1;Replay 1
Val	Phone 3	21-35	Living 1-3; Print 1,2; Replay 1,2
Test	Phone 1,2,4,5,6	36-55	Living 1; Print 2; Replay 2

Table 2: Zero- and few-shot FAS benchmark: *Cross-ZF*.

Set	Domains	PSAI
Train	CASIA-MFSD, MSU-MFSD, SiW	Living 1-7; Print 1-4; Replay 1-7
Val	MSU-USSA	Living 1;Print 1-2; Replay 1-6
Test	3DMAD, Oulu-NPU, Replay-Attack	Living 1-9; 3D Mask; Print 1-3;Replay 1-4

Table 3: Zero- and few-shot FAS benchmark: *SURF-ZF*.

Set	Modals	PSAI
Train	RGB;Depth	Living 1;Print 1;Cut 1-5
Val	IR;Depth	Living 1;Print 1;Cut 1-5
Test	IR:Depth	Living 1;Print 1;Cut 1-5

σ is the standard deviation of $ACER$ on all the test tasks, and T is the quatity of test tasks.

Implementation Details In our experiment, we generate the ground-truth depth label of living face with the PRNet (Feng et al. 2017), and normalize the generated facial depth to [0, 1]. To distinguish spoofing face from living face, we set the ground-truth depth label of spoofing face to all zero. The generated facial depths are shown in Fig. 4. All the facial depth maps are resized into 32×32 resolution.

We generate 100,000 training tasks on the train set and 100 ($T = 100$) testing tasks on the test set. For each K-shot training task, K is randomly sampled from $\{0,1,3,5,7,9\}$. For testing tasks, K is a specified number indicating the meta-learner is tested on specified K-shot tasks. For example, if we evaluate the meta-learner's performance on zero-shot FAS tasks, we set $K = 0$ and generate 100 such zero-shot testing tasks to test the meta-learner. We set Q to 15, M to 10. The meta batch size is set to 8, and the meta-learning rate β is set to 0.0001. The AIU parameters α and γ are initialized to 0.001 and 1, respectively.

Compared Methods To validate the performance of AIM-FAS on zero- and few-shot FAS problem, we compare AIM-FAS with three FAS detectors Resnet-10, FAS-DR, and DTN*. The detector FAS-DR is the network of AIM-FAS trained in traditional supervised learning. As the network of detector FAS-DR is the same as that of AIM-FAS, We treat detector FAS-DR as the baseline of AIM-FAS. The detector Resnet-10 is a binary classification FAS model and is also trained in traditional supervised learning. DTN(Liu et al. 2019) is a zero-shot FAS detector. We re-implement DTN with all experiment settings the same to the original paper and named it as DTN*. For fairly comparison, we set up the evaluation protocol for all methods, which is shown in Tab.4. For example, the detector Resnet-10 is trained on the train set, and to evaluate its 0-shot performance, we evaluate it directly on the query set of 0-shot FAS tasks without finetuning on the support set. To evaluate its 1-shot performance, we first finetune it on the support set of the 1-shot tasks and then evaluate it on the corresponding query set.

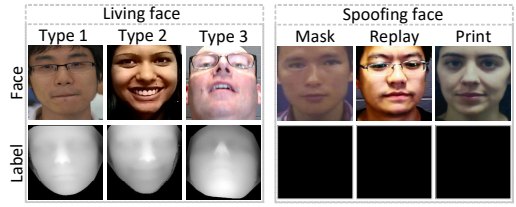

Figure 4: Generated depth label of living and spoofing faces.

Table 4: Evaluation detail of compared methods and AIM-FAS.

Method	Train	0-shot Test		1- or 5-shot Test	
		Finetune	Evaluate	Finetune	Evaluate
Compared	Train set	/	Query	Support	Query
AIM-FAS	Training tasks	Support	Query	Support	Query

5.2 Experiment on Proposed Benchmarks

Corresponding experimental results on the proposed benchmarks are shown in Tab.5. It can be seen that AIM-FAS outperforms the other detectors with a clear margin on all benchmarks. Note that, as original DTN is designed for zero-shot FAS, we follow the same way to evaluate DTN* on zero-shot instead of few-shot tasks. Compared with FAS-DR, the *ACER* of AIM-FAS decreases by 25%, 17%, and 28% on zero-, 1- and 5-shot tasks on *OULU-ZF*, respectively, and decreases by 38%, 30%, and 38% on *Cross-ZF*, and decreases by 12%, 13%, and 16% on *SURF-ZF*. Since AIM-FAS trains the meta-learner to focus on learning discrimination of new spoofing types from predefined faces, or from predefined faces and a few examples of new living and spoofing types. It learns more generalized discriminative features for detecting new attack types. Whereas, FAS-DR only focus on learning the discrimination to distinguish predefined spoofing faces from living faces.

Another phenomenon is that the margin between AIM-FAS and the other methods is more clear on *Cross-ZF* than on the other benchmarks. The possible reason behind is that *Cross-ZF* contains more diverse fine-grained living and spoofing categories, which is more suitable than the other benchmarks for AIM-FAS learning general discrimination for detecting new attack types.

5.3 Experiment on Existing Dataset

To further evaluate the advantages of AIM-FAS, we test AIM-FAS on the protocol proposed by (Arashloo, Kittler, and Christmas 2017). In this protocol, CASIA, Replay-Attack, and MSU-MFSD are used to evaluate the FAS model's zero-shot performance across replay and print attacks. As shown in Tab.6, AIM-FAS performs better than the other methods on most sub-protocols with rising the average AUC by at least 0.52%. The result further reveals that AIM-FAS is successful for not only few-shot but also zero-shot FAS. AIM-FAS performs not the best on the sub-protocols of CASIA Video, Replay-Attack Video, and MSU Printed Photo. The possible reason is that the training spoofing categories of these sub-protocols are unitary and not suitable for AIM-FAS learning the discrimination to detect the testing spoofing category. For example, on CASIA, the Cut Photo and Warped Photo spoofing categories are not varied enough, and the meta-learner trained on these categories can hardly summarize and capture the general discrimination that is effective for detecting the Video category.

Table 5: Experimental result on three proposed benchmarks. DTN* is our re-implementation of DTN(Liu et al. 2019) with the same setting as its original paper.

Benchmark	Method	ACER(%)		
		0-shot	1-shot	5-shot
OULU-ZF	Resnet-10	7.27±1.42	7.13±1.19	4.20±1.12
	DTN*	5.83±1.16	/	/
	FAS-DR	6.60±1.78	4.83±1.40	3.37±1.02
	AIM-FAS	**4.97±1.29**	**4.00±1.31**	**2.44±0.71**
Cross-ZF	Resnet-10	26.51±2.59	26.37±2.64	15.43±2.38
	DTN*	17.62±3.89	/	/
	FAS-DR	13.49±3.62	10.54±3.03	7.21±1.87
	AIM-FAS	**8.43±2.92**	**7.34±1.45**	**3.11±1.01**
SURF-ZF	Resnet-10	45.60±0.72	45.43±0.81	44.53±1.40
	DTN*	45.27±2.29	/	/
	FAS-DR	34.61±2.15	33.17±2.07	32.50±1.51
	AIM-FAS	**30.97 ±1.28**	**28.75 ±1.49**	**27.27±1.25**

5.4 Ablation Study

AIM-FAS for Binary Supervision We validate the effectiveness of AIM-FAS on binary supervised architecture by taking Resnet-10 as the backbone, named as AIM-FAS (Resnet). And Resnet-10 trained in traditional supervised manner is set as the baseline of AIM-FAS (Resnet). The comparison of these two methods on *Cross-ZF* is shown in Tab.7. Compared with Resnet-10, AIM-FAS (Resnet) decrease the ACER by 45.08%, 54.72%, and 41.55% on 0-shot, 1-shot, and 5-shot tasks, respectively. This demonstrates the generality of AIM-FAS on different network structures and different supervision manners.

Effectiveness of Predefined Living and Spoofing Faces in Support Set Here we verify whether predefined living and spoofing faces in the support set are useful for AIM-FAS to learn discrimination to detect the new spoofing category. In this experiment, during the testing stage, we generate K-shot FAS tasks without predefined living and spoofing categories in the support set. In other words, the support set of K-shot FAS tasks here contains no categories L_i and S_m in Fig.2(b). For K-shot ($K > 0$) FAS tasks, AIM-FAS without PreDefined living and spoofing faces (named as AIM-FAS w/o PD) inner-updates the meta-learner with only $2K$ new type of spoofing/living faces, and then test the meta-learner on the query set. Note that we do not test AIM-FAS w/o PD on zero-shot FAS tasks since that support set of zero-shot FAS tasks here is empty. In Tab.7, we can see that AIM-FAS w/o PD increases the ACER(%) by 11% and 42% on 1-shot and 5-shot FAS, respectively. The worse performance of AIM-FAS w/o PD indicates that the predefined living and spoofing faces indeed bring benefit for the meta-learner learning discrimination for detecting new attacks.

Effectiveness of Fusion Training (FT) For the trained meta-learner to be capable of solving both zero- and few-shot FAS problems, we present an FT strategy in AIM-FAS for training the meta-learner simultaneously on all K-shot

Table 6: Performance of AIM-FAS on CASIA, Replay-Attack, and MSU-MFSD. The evaluation metric is AUC(%).

Methods	CASIA			Replay-Attack			MSU			Overall
	Video	Cut Photo	Warped Photo	Video	Digital Photo	Printed Photo	Printed Photo	HR Video	Mobile Video	
OC-SVM_RBF+BSIF	70.7	60.7	95.9	84.3	88.1	73.7	64.8	87.4	74.7	78.7±11.7
SVM_RBF+BSIF	91.5	91.7	84.5	99.1	98.2	87.3	47.7	99.5	97.6	88.6±16.3
NN+LBP	**94.2**	88.4	79.9	99.8	95.2	78.9	50.6	99.9	93.5	86.7±15.6
DTN	90.0	97.3	97.5	**99.9**	99.9	99.6	**81.6**	99.9	97.5	95.9±6.2
AIM-FAS(ours)	93.6	**99.7**	**99.1**	99.8	**99.9**	**99.8**	76.3	**99.9**	**99.1**	**96.4±7.8**

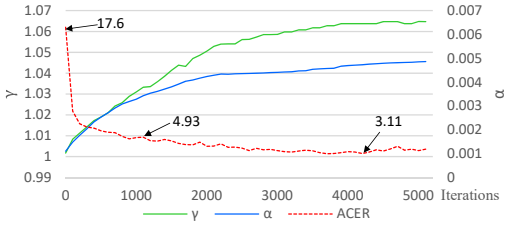

Figure 5: The learning curve of α, γ and the meta-learner's ACER. The left Y-axis is the Y-axis of γ. The right Y-axis is the Y-axis of α. The X-axis is the training iteration.

Table 7: Ablation experiment on *Cross-ZF*.

Method	ACER (%)		
	0-shot	1-shot	5-shot
Resnet-10	26.51±2.59	26.37±2.64	15.43±2.38
AIM-FAS(Resnet)	**14.56±2.63**	**11.94±1.85**	**9.02±1.69**
AIM-FAS w/o AIU	11.67±2.08	11.53±2.96	6.44±1.57
AIM-FAS w/o FT	12.61±1.67	9.15±1.73	3.25±1.05
AIM-FAS w/o PD	/	8.25±1.16	7.66±2.45
AIM-FAS	**8.43±2.92**	**7.34±1.45**	**3.11±1.01**

tasks. To assess FT, we conduct an experiment that trains the meta-learner without FT. AIM-FAS w/o FT trains a meta-learner on 0-shot tasks for 0-shot testing, and train another meta-learner on 1-shot tasks for 1-shot testing, and so on. Tab.7 shows the performance of AIM-FAS w/o FT. Compared with AIM-FAS w/o FT, the AIM-FAS performs better on all kinds of shot scenes. The possible reason is that the FT manner provides the meta-learner diverse K-shot FAS tasks so that the AIM-FAS meta-learner generalizes better from training tasks to testing tasks. In other words, with FT, the testing shot scene is a subset of the training shot scenes.

Impact of Adaptive Inner-Update (AIU) In this experiment, we discard the Adaptive Inner-Update (AIU) from the complete AIM-FAS. Tab.7 shows the comparison of AIM-FAS and AIM-FAS w/o AIU. We find that the AIU improves our AIM-FAS with a large margin. Specifically, AIM-FAS with AIU apparently decreases ACER(%) more than 3.0 on 0-, 1- and 5-shot. Furthermore, in Fig.5, we show the curves of α and γ of Eq. 2 during meta-training process. Both α and γ present a rising tendency, meanwhile the ACER falls down. This indicates that AIM-FAS prefers a larger inner-update learning rate, and with the learned α and γ, AIM-FAS performs better than the AIM-FAS w/o AIU.

5.5 Visualization and Analysis

In this subsection, the feature (feature of the last but one layer) distribution of the meta-learner is illustrated in Fig.6.

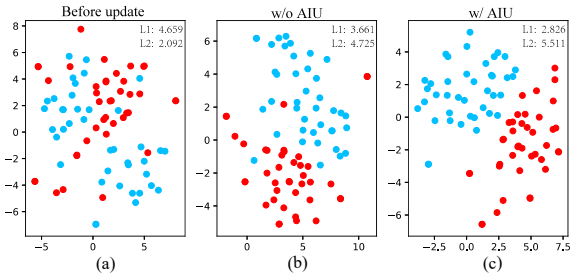

Figure 6: Visualization of the distribution of living and spoofing faces in the query set of a 5-shot FAS testing task. Color used: *red*=living, *blue*=spoofing.

We randomly generate a 5-shot FAS testing task and update the meta-learner for 50 inner-update steps on the support set. Then the feature distribution of the query set is visualized with t-SNE(Maaten and Hinton 2008). Fig.6a is the feature distribution of the query set before the meta-learner updates itself on the support set. Fig.6b and Fig.6c are the distributions after the meta-learner updates itself for 50 inner-update steps without and with AIU, respectively. We also show the category inner distance (L1) and the inter distance (L2). From left to right, the distinction between the distribution of living and spoofing faces turns more and more clear, and L1 declines gradually, whereas L2 rises. The visualization clearly reveals that the meta-learner learns the discrimination between new living and spoofing categories on the support set, and the AIU helps the meta-learner to learn better discrimination.

6 Conclusion and Future Work

In this paper, we redefine the face anti-spoofing (FAS) as a simultaneously zero- and few-shot learning issue. To address this issue, we develop a novel method Adaptive Inner-update Meta Face Anti-Spoofing (AIM-FAS) and propose three zero- and few-shot FAS benchmarks. To validate AIM-FAS, we conduct experiments on both the proposed benchmarks and existing zero-shot protocols. All experiments show that AIM-FAS outperforms existing methods with a clear margin on both zero- and few-shot FAS. In the future, we will develop AIM-FAS to more challenging and practical application scenes.

7 Acknowledgments

This work was supported by the Chinese National Natural Science Foundation Projects (Grant No. 61876178, 61872367, and 61806196).

References

Arashloo, S. R.; Kittler, J.; and Christmas, W. J. 2017. An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol. *IEEE Access* 5:13868–13882.

Atoum, Y.; Liu, Y.; Jourabloo, A.; and Liu, X. 2017. Face anti-spoofing using patch and depth-based cnns. In *IJCB*, 319–328.

Bengio, S.; Bengio, Y.; Cloutier, J.; and Gecsei, J. 1992. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, 6–8. Univ. of Texas.

Boulkenafet, Z.; Komulainen, J.; and Hadid, A. 2016. Face spoofing detection using colour texture analysis. *IEEE Transactions on Information Forensics and Security* 11(8):1818–1830.

Boulkenafet, Z.; Komulainen, J.; and Hadid, A. 2017. Face antispoofing using speeded-up robust features and fisher vector encoding. *IEEE Signal Processing Letters* 24(2):141–145.

de Freitas Pereira, T.; Anjos, A.; De Martino, J. M.; and Marcel, S. 2012. Lbp- top based countermeasure against face spoofing attacks. In *ACCV*, 121–132.

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

Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P. L.; Sutskever, I.; and Abbeel, P. 2016. Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Feng, Y.; Wu, F.; Shao, X.; Wang, Y.; and Zhou, X. 2017. Joint 3d face reconstruction and dense alignment with position map regression network. In *CVPR*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*.

Gan, J.; Li, S.; Zhai, Y.; and Liu, C. 2017. 3d convolutional neural network based on face anti-spoofing. In *ICMIP*, 1–5.

Grant, E.; Finn, C.; Levine, S.; Darrell, T.; and Griffiths, T. 2018. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*.

international organization for standardization. 2016. Iso/iec jtc 1/sc 37 biometrics: Information technology biometric presentation attack detection part 1: Framework. In *https://www.iso.org/obp/ui/iso*.

Komulainen, J.; Hadid, A.; and Pietikainen, M. 2013. Context based face anti-spoofing. In *BTAS*, 1–8.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2009. Learning to detect unseen object classes by between-class attribute transfer. 951–958.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(3):453–465.

Li, L.; Feng, X.; Boulkenafet, Z.; Xia, Z.; Li, M.; and Hadid, A. 2016. An original face anti-spoofing approach using partial convolutional neural network. In *IPTA*, 1–6.

Liu, Y.; Stehouwer, J.; Jourabloo, A.; and Liu, X. 2019. Deep tree learning for zero-shot face anti-spoofing.

Liu, Y.; Jourabloo, A.; and Liu, X. 2018. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *CVPR*, 389–398.

Lucena, O.; Junior, A.; Moia, V.; Souza, R.; Valle, E.; and Lotufo, R. 2017. Transfer learning using convolutional neural networks for face anti-spoofing. In *International Conference Image Analysis and Recognition*, 27–34.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of machine learning research* 9(Nov):2579–2605.

Määttä, J.; Hadid, A.; and Pietikäinen, M. 2011. Face spoofing detection from single images using micro-texture analysis. In *IJCB*, 1–7.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2018. A simple neural attentive meta-learner.

Nagpal, C., and Dubey, S. R. 2018. A performance evaluation of convolutional neural networks for face anti spoofing. *arXiv preprint arXiv:1805.04176*.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On first-order meta-learning algorithms.

Norouzi, M.; Mikolov, T.; Bengio, S.; Singer, Y.; Shlens, J.; Frome, A.; Corrado, G.; and Dean, J. A. 2013. Zero-shot learning by convex combination of semantic embeddings. *arXiv: Learning*.

Patel, K.; Han, H.; and Jain, A. K. 2016a. Cross-database face antispoofing with robust feature representation. In *Chinese Conference on Biometric Recognition*, 611–619.

Patel, K.; Han, H.; and Jain, A. K. 2016b. Secure face unlock: Spoof detection on smartphones. *IEEE transactions on information forensics and security* 11(10):2268–2283.

Qin, Y.; Zhang, W.; Zhao, C.; Wang, Z.; Shi, H.; Qi, G.; Shi, J.; and Lei, Z. 2018. Rethink and redesign meta learning. *CoRR* abs/1812.04955.

Shao, R.; Lan, X.; and Yuen, P. C. 2017. Deep convolutional dynamic texture learning with adaptive channel-discriminability for 3d mask face anti-spoofing. In *IJCB*, 748–755.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 4077–4087.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 3630–3638.

Wang, Z.; Zhao, C.; Qin, Y.; Zhou, Q.; and Lei, Z. 2018. Exploiting temporal and depth information for multi-frame face anti-spoofing. *arXiv preprint arXiv:1811.05118*.

Xu, Z.; Li, S.; and Deng, W. 2015. Learning temporal features using lstm-cnn architecture for face anti-spoofing. In *ACPR*, 141–145.