Cross-Domain Face Presentation Attack Detection Techniques with Attention to Genuine Faces

Li Zhi

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

08/04/2023

Date

Li Zhi

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

08/04/2023

Date

Prof. Lam Kwok Yan

Authorship Attribution Statement

This thesis contains material from 2 papers published in the following peer-reviewed journals and 1 paper accepted at conferences in which I am listed as an author.

Chapter 3 is published as Z. Li, H. Li, K. -Y. Lam and A. C. Kot, "Unseen Face Presentation Attack Detection with Hypersphere Loss," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 2852-2856, doi: 10.1109/ICASSP40776.2020.9054420.

The contributions of the co-authors are as follows:

- Prof. Alex C. Kot and Prof. Kwok-Yan Lam suggested the topic.
- I designed the initial method and the method was improved by discussions with Assist/Prof. Haoliang Li, Prof. Kwok-Yan Lam, and Prof. Alex C. Kot.
- I implemented the proposed method.
- I designed and conducted experiments with the suggestions provided by Assist/Prof. Haoliang Li.
- I wrote the initial manuscript. Assist/Prof. Haoliang Li, Prof. Kwok-Yan Lam, and Prof. Alex C. Kot revised the manuscript.

Chapter 4 is published as Z. Li, H. Li, X. Luo, Y. Hu, K. -Y. Lam and A. C. Kot, "Asymmetric Modality Translation For Face Presentation Attack Detection," in IEEE Transactions on Multimedia, doi: 10.1109/TMM.2021.3121140.

The contributions of the co-authors are as follows:

- Prof. Alex C. Kot and Prof. Kwok-Yan Lam suggested the topic.
- I designed the initial method and the method was improved by discussions with Assist/Prof. Haoliang Li, Prof. Yongjian Hu, Prof. Kwok-Yan Lam, and Prof. Alex C. Kot.
- I implemented the proposed method with the assistance of Mr. Xin Luo.
- I designed and conducted the experiments with the suggestions provided by Assist/Prof. Haoliang Li.
- I wrote the initial manuscript. Assist/Prof. Haoliang Li, Mr. Xin Luo, Prof. Yongjian Hu, Prof. Kwok-Yan Lam, and Prof. Alex C. Kot revised the manuscript.

Chapter 5 is published as Z. Li, R. Cai, H. Li, K. -Y. Lam, Y. Hu and A. C. Kot, "One-Class Knowledge Distillation for Face Presentation Attack Detection," in IEEE Transactions on Information Forensics and Security, doi: 10.1109/TIFS.2022.3178240.

The contributions of the co-authors are as follows:

- Prof. Alex C. Kot and Prof. Kwok-Yan Lam suggested the topic.
- I designed the initial method and the method was improved by discussions with Mr. Rizhao Cai, Assist/Prof. Haoliang Li, Prof. Yongjian Hu, Prof. Kwok-Yan Lam, and Prof. Alex C. Kot.
- I implemented the proposed method.
- I devised the evaluation protocols with suggestions provided by Assist/Prof. Haoliang Li.
- I designed and conducted experiments with the assistance of Mr. Rizhao Cai.
- I wrote the initial manuscript. Mr. Rizhao Cai, Assist/Prof. Haoliang Li, Prof. Yongjian Hu, Prof. Kwok-Yan Lam, and Prof. Alex C. Kot revised the manuscript.

08/04/2023

. . . . . .

Date

. . . . . .

Li Zhi

Acknowledgements

In the summer of 2018, I started my Ph.D. adventure with longings and curiosities for scientific research. Four years passed quickly, faster than the end of COVID-19, but it is not an easy journey for me. I sincerely thank the people I met during the journey for helping and accompanying me through this important period of my life.

First of all, I would like to express my greatest gratitude to my supervisors Prof. Kwok-Yan Lam and Prof. Alex C. Kot. I thank them for giving me the opportunity to study at Nanyang Technological University, Singapore, under their supervision. Moreover, I am grateful that they always provide me with careful guidance and support for my research work. Their enthusiasm and attitude towards work and life have also deeply influenced me.

My Ph.D. journey is not always smooth and I was once lost in my research. I sincerely thank Prof. Kot for enlightening me during the most difficult time of my Ph.D. study. I may not be able to complete the journey without his help.

I would like to thank Prof. Haoliang Li for his guidance in my research. Before starting my Ph.D. programme, I didn't have much experience in research. I thank Haoliang for his patient guidance and advice. I did benefit a lot from the discussion with him. Besides, I want to thank my TAC members Prof. Shijian Lu and Prof. Bihan Wen for their efforts in reviewing my research progress and providing advice. I also want to thank Prof. Yongjian Hu for his help in my research work.

I am so glad to meet and work with my lovely colleagues. I would like to thank Renjie Wan, Jun Liu, Yi Huang, Jiale Guo, Wenzhuo Yang, Ziyao Liu, Chen Chen, Jiani Fan, Siyuan Yang, Rizhao Cai, Chenyu Yi, Ling Li, Yufei Wang, Yi Yu, Shan Lin, Wenhan Yang, Zitong Yu, Rongkai Zhang, Hao Cheng, Lanqing Guo, Zhan Lu, and others in ROSE Lab and SPIRIT Lab for their accompanying. I did enjoy the time spent with them.

I also appreciate Dr. Dennis Sng, Ms. Qian Wang, Ms. Ooy Mei Chai, Ms. Angela Ho, and the staff in charge of SCSE graduate student affairs for their administrative assistance and help.

Lastly, I want to thank my family and friends. I thank my parents, my sister, and my little niece for their always support and love. I thank my friends Ning Zhang, Wenjun Li, Hang Li, Gaochen Shi, and Zehao Liu for encouraging me and helping me release stress.

Contents

Acknowledgements ix

List of Figures xv

List of Tables xix

Symbols xxi

Abbreviations xxv

Summary xxix

1 Introduction 1
- 1.1 Background of Face Presentation Attack Detection . . . 1
- 1.2 Major Contributions . . . 2
- 1.3 Organization of the Thesis . . . 4

2 Literature Review 5
- 2.1 Single VIS-Modality Face PAD . . . 6
  - 2.1.1 Conventional Methods . . . 6
    - 2.1.1.1 Feature Representations . . . 6
    - 2.1.1.2 Classifiers . . . 7
  - 2.1.2 Deep Learning Methods . . . 7
    - 2.1.2.1 Network Architectures . . . 8
    - 2.1.2.2 Auxiliary Tasks . . . 8
    - 2.1.2.3 Frequency Domain Analysis . . . 9
    - 2.1.2.4 Domain Generalization Techniques . . . 9
    - 2.1.2.5 Domain Adaptation Techniques . . . 11
- 2.2 Multi-Modality Face PAD . . . 12
- 2.3 Datasets and Evaluation Metrics . . . 13
  - 2.3.1 Single VIS-Modality Datasets . . . 13
  - 2.3.2 Multi-Modality Datasets . . . 15
  - 2.3.3 Evaluation Metrics . . . 16

3 Unseen Face PAD with Hypersphere Loss **19**
3.1 Introduction . . . 19
3.2 Methodology . . . 20
3.2.1 Hypersphere Loss Function . . . 21
3.2.2 Framework of the Proposed Method . . . 24
3.3 Experimental Setup . . . 25
3.3.1 Dataset Information . . . 25
3.3.2 Evaluation Protocols . . . 25
3.3.3 Evaluation Metrics . . . 25
3.3.4 Baseline Methods . . . 25
3.3.5 Implementation Details . . . 26
3.4 Results and Analysis . . . 27
3.4.1 Unseen-Attack Experiments . . . 27
3.4.2 Discussion . . . 29
3.5 Chapter Summary . . . 31

4 Asymmetric Modality Translation for Face PAD **33**
4.1 Introduction . . . 33
4.2 Methodology . . . 35
4.2.1 Framework of the Proposed Method . . . 35
4.2.2 Asymmetric Modality Translation . . . 36
4.2.3 Modality Fusion and Discrimination . . . 38
4.2.4 PLGF-based Illumination Normalization . . . 40
4.3 Experimental Setup . . . 42
4.3.1 Dataset Information . . . 42
4.3.2 Evaluation Protocols . . . 43
4.3.3 Evaluation Metrics . . . 44
4.3.4 Baseline Methods . . . 44
4.3.5 Implementation Details . . . 45
4.4 Results and Analysis . . . 46
4.4.1 Grand-Test Experiments . . . 46
4.4.2 Cross-Illumination Experiments . . . 48
4.4.3 Unseen-Attack Experiments . . . 49
4.4.4 Cross-Dataset Experiments . . . 51
4.4.5 Ablation Study . . . 52
4.4.6 Discussions . . . 54
4.5 Chapter Summary . . . 58

5 One-Class Knowledge Distillation for Face PAD **59**
5.1 Introduction . . . 59
5.2 Methodology . . . 62
5.2.1 Problem Formulation . . . 62
5.2.2 Framework of the Proposed Method . . . 63

5.2.3 Discriminative Teacher Stream . . . 63
5.2.4 Stubborn Student Stream . . . 65
5.2.5 Inference at the Test Phase . . . 69
5.3 Experimental Setup . . . 69
5.3.1 Dataset Information . . . 69
5.3.2 Evaluation Protocols . . . 70
5.3.3 Evaluation Metrics . . . 71
5.3.4 Baseline Methods . . . 71
5.3.5 Implementation Details . . . 72
5.4 Results and Analysis . . . 73
5.4.1 General One-Class Domain Adaptation Experiments . . . 73
5.4.2 Client-Specific One-Class Domain Adaptation Experiments . 76
5.4.3 Ablation Study . . . 79
5.4.4 Discussion . . . 83
5.5 Chapter Summary . . . 84

6 Conclusion and Future Work **85**
6.1 Conclusion . . . 85
6.2 Future Work . . . 87

List of Author's Awards, Patents, and Publications **89**

Bibliography **91**

List of Figures

3.1 The figure illustrates the target feature space and the proposed hypersphere loss. As shown in figure (A), the genuine face samples are expected to converge to a small hypersphere of radius r while the attack samples should be separated from the smaller hypersphere by a margin m. The figure (B) illustrates the distances in the proposed hypersphere loss. 22

3.2 The figure illustrates the framework of the proposed method. The feature extractor is implemented with ResNet18, which embeds the preprocessed face image into a 128 dimensional feature space. The training of the feature extractor is directed by minimizing the hypersphere loss. At the testing phase, the squared L_2 norm of the feature vector is used as the score compared with the predetermined threshold for inference. 24

3.3 The figure illustrates the overall performance of our method on SiW-M dataset. The average ACER, average APCER, and average BPCER of 13 sub-experiments are plotted in blue, yellow, and green colors. The horizontal and vertical axes represent the threshold and the value of evaluation metrics, respectively. 30

3.4 The figure illustrates the specific performance of our method on SiW-M dataset. The ACER of the 13 sub-experiments are plotted in different colors. The horizontal and vertical axes represent the threshold and the value of ACER, respectively. 30

4.1 The figure illustrates the concept of asymmetric modality translation. We expect the modality translator can successfully translate genuine face images to the target modality while fails for attacks. . 34

4.2 The figure illustrates the framework of our proposed method, which consists of three modules. The Illumination Normalization (IN) module selectively reduces the impact of illumination variations on sensitive modalities. The Asymmetric Modality Translation (AMT) module translates the source modality image to the target modality. The Discrimination (DC) module fuses the translated image with the ground-truth target modality image as the input for inference. . 36

4.3 The figure shows the illustration of two fusion operations. F_{concat} and $F_{subtract}$ denote the concatenation and subtraction operation respectively. W and H denote the width and height of the image. . 39

4.4 The figure shows the visualization results of illumination normalization. The first row of (A) shows the raw VIS images of genuine faces in MSSPOOF dataset that captured under different illumination conditions. The second row of (A) shows corresponding VIS images processed by IN module (IN-VIS). Sub-figure (B) shows VIS and IN-VIS images of attack samples. 41
4.5 The figure shows the visualization of images under WMCA (T-I) setting. Columns (A)-(C) are images of genuine face samples, and columns (D)-(H) are images of attack samples. From top to bottom row are images of ground-truth IN-NIR, IN-NIR that translated from thermal images and corresponding patch maps. The patch maps are colorized as heat maps, where the red region indicates anomaly. 46
4.6 The figure shows the visualization of images under WMCA (T-I) setting. Column (A) are images of a same genuine face sample, and columns (B)-(F) are images of different types of attacks. From top to bottom row are images of ground-truth thermal, ground-truth NIR, NIR that translated from thermal, ground-truth IN-NIR and IN-NIR that translated from thermal images. Noted that samples of glasses attack and fake head attack are not visualized here because no samples are in authorized list due to the privacy issue. 54
4.7 The figure shows the visualization of images under WMCA (T-I) setting. Column (A) are images of a same genuine face sample from WMCA, and columns (B)-(F) are images of different attacks. From top to bottom row are images of ground-truth IN-NIR, translated IN-NIR without and with the translation block. 55
4.8 Visualization of Receiver Operating Characteristic (ROC) curves for WMCA(T-I) grand-test evaluation. The horizontal axis represents the False Detection Rate (FDR) and the vertical axis represents the True Detection Rate (TDR). 57

5.1 The figure illustrates the framework under the general setting. It contains a teacher network trained with the source domain data to provide discriminative features, and a student network trained with the target domain genuine face data to generate similar features to the teacher's descriptions. In the test phase, the face images will be fed into both the DT and SS netorks for feature extraction and the similarity between features of the two networks will be used as the inference score. 61
5.2 The figure illustrates the framework under the client-specific setting. After the client-specific one-class domain adaptation, the framework contains a teacher network and a set of N student networks. Each student network serves for one specific target client. In the test phase, the face images will be fed into the DT network and the corresponding SS network for client-specific inference. 61

5.3 The figure illustrates the network training of the Discriminative Teacher network with the source domain data. The face image x_i will be fed into several convolutional blocks to extract features f_i^1, f_i^2 and f_i^3. The multi-level features are fed into the final convolutional block for pixel map d_i estimation. 65
5.4 The figure illustrates the training of the Stubborn Student network. The target domain genuine face images z_i will be fed into both the teacher and student networks for feature extraction. The multi-level similarities between the DT and SS networks are computed to guide the optimization of the SS network. During the training, a sparse training strategy is used to reduce the parameter density of the SS network. 65
5.5 The figure shows the score distributions of the DT on 10 tasks of C-N-CS protocol. A pair of blue and yellow boxes are used to illustrate the score distribution for each task. The blue and yellow boxes illustrate the distributions of the testing genuine face and attack samples, respectively. 78
5.6 The figure shows the score distributions of different methods on C-N-CS task 1. From the left to the right are score distributions of the DT, DT+finetune, DT+GMM, DT+OCSVM and our proposed method. The blue and yellow boxes illustrate the distributions of the genuine face and attack samples, respectively. The figure shows that our method has larger separation gap compared to baseline methods. 78
5.7 The figure shows the Receiver Operating Characteristic (ROC) Curves of different methods on C-N-CS task 1. The horizontal axis is the False Detection Rate (FDR) and the vertical axis is the True Detection Rate (TDR). 79
5.8 This figure shows the average HTER of our proposed method over the number of training iterations on C-N-CS tasks. The four curves illustrate the impact of the regrowth mechanism on the proposed method at two SS model density conditions. 81
5.9 The figure (a), (b), (c) show the average HTER with different level features pretrained on I, M, C dataset, respectively. The average HTER performance with Level 1, Level 2, and Level 3 features are plotted in blue, green, and yellow, respectively. 82
5.10 The figures show examples of the image samples correctly classified and misclassified by our model on C-N-CS task 1. Figure (A) shows the correctly classified genuine face samples; figure (B) shows the misclassified genuine face samples; figure (C) shows the correctly classified attack samples; figure (D) shows the misclassified attack samples. 83

List of Tables

3.1 Unseen-Attack Experimental Results AUC(%) on CASIA-FASD, IDIAP REPLAY-ATTACK, and MSU-MFSD Datasets 28
3.2 Unseen-Attack Experimental Results on SiW-M 28

4.1 List of Dataset Information 42
4.2 Grand-Test Experimental Results on WMCA (Same Modality) . . . 47
4.3 Grand-Test Experimental Results on WMCA (Different Modality) . 47
4.4 Grand-Test Experimental Results on CASIA-SURF 48
4.5 Grand-Test Experimental Results on MSSPOOF 48
4.6 Cross-Illumination (LOO) Experimental Results on WMCA 49
4.7 Unseen-Attack (LOO) Experimental Results on WMCA (Same Modality) 49
4.8 Unseen-Attack (LOO) Experimental Results on WMCA (Different Modality) 50
4.9 Unseen-Attack (LOO) Experimental Results On CASIA-SURF . . . 50
4.10 Unseen-Attack (LTO) Experimental Results on CASIA-SURF . . . 51
4.11 Cross-Dataset Experimental Results 51
4.12 Experimental Results with Different Components 53
4.13 Experimental Results with Different Fusion Operations 53
4.14 Experimental Results with/without Translation Blocks 55
4.15 Experimental Results with Illumination Normalization Module . . . 55
4.16 Grand-Test Experimental Results on WMCA (Score Fusion) 56
4.17 Grand-Test Experimental Results on CASIA-SURF (Score Fusion) . 57

5.1 Performance Comparison with the One-Class Domain Adaptation Methods on the CIMN-OCDA Protocol (ideal experimental setting) 74
5.2 Performance Comparison with the One-Class Domain Adaptation Methods on the CIMN-OCDA Protocol (challenging experimental setting) 74
5.3 Performance Comparison with the Unsupervised Domain Adaptation Methods 74
5.4 Performance Comparison with Baseline Methods on the OULU-OCA-SA Protocol 75
5.5 Performance (HTER) Comparison with the One-class Domain Adaptation Methods on the WMCA-CI-OCDA Protocol 75

5.6 Performance (AUC) Comparison with the One-class Domain Adaptation Methods on the WMCA-CI-OCDA Protocol 75
5.7 Performance (HTER) Comparison with the One-class Domain Adaptation Methods on the CIM-N-CS-OCDA Protocol 76
5.8 Performance (AUC) Comparison with the One-class Domain Adaptation Methods on the CIM-N-CS-OCDA Protocol 76
5.9 Performance of the Proposed Method with Different Stubborn Student Density on the CIM-N-CS-OCDA Protocol 80
5.10 Model Size of the Stubborn Student at Different Density 80
5.11 Performance of the Proposed Method with Single-Level and Multi-Level Distillation on the CIM-N-CS-OCDA Protocol 80

Symbols

Symbols in Chapter 2

N_B	the number of bona-fide samples
N_{PAI}	the number of attack samples in specific species
RES_i	an indicator takes value 1 if the ith sample is classified as an attack.

Symbols in Chapter 3

L_2	the Euclidean distance
N_a	the number of attack samples
N_g	the number of genuine face samples
$\mathcal{L}_a$	the loss of attack samples
$\mathcal{L}_g$	the loss of genuine face samples
$\mathcal{L}_h$	the total hypersphere loss
d_i^a	the distance of the ith attack sample
d_i^g	the distance of the ith genuine face sample
d^s	the distance of the smaller hypersphere
d^l	the distance of the larger hypersphere
r	the radius of the smaller hypersphere
m	the margin from the smaller to the larger hypersphere
λ_a	the weight of attack samples loss
λ_g	the weight of genuine face samples loss
$\|\cdot\|_2$	the L_2 norm
$\langle\cdot,\cdot\rangle$	the inner product of two vectors

Symbols in Chapter 4

$\mathcal{L}_{AMT}$	the asymmetric modality translation loss
$\mathcal{L}_{dis}$	the discrimination loss
$\mathcal{L}_{pixel}$	the pixel-level loss
$\mathcal{L}_{latent}$	the latent-level loss
$\mathcal{L}_{latent,i}$	the latent-level loss of i-th data sample
$\mathcal{L}_{total}$	the total loss
τ	constant hyperparameter
λ_1	the weight of pixel-level loss
λ_2	the weight of latent-level loss
λ_3	the weight of discrimination loss
r_i	the raw image of the i-th data sample
r_i^S	the raw source modality image of the i-th data sample
r_i^T	the raw target modality image of the i-th data sample
x_i	the IN-processed image of the i-th data sample
x_i^S	the IN-processed source modality image of the i-th data sample
x_i^T	the IN-processed target modality image of the i-th data sample
$x_i'^T$	the translated target modality image of the i-th data sample
f_i	the fusion results of the i-th data sample
y_i	the label of the i-th data sample
d_i	the estimated pixel map of the i-th data sample
i	the index of data sample
g	the index of genuine face sample
a	the index of attack sample
z_i	the latent representation of the i-th data sample
z_g	the latent representation of the g-th genuine face sample
z_a	the latent representation of the a-th attack sample
G	the set of genuine face samples
A	the set of attack samples
N_g	the number of genuine face samples

W	the width of the image
H	the height of the image
M_x	the horizontal filter mask of PLGF descriptor
M_y	the vertical filter mask of PLGF descriptor

Symbols in Chapter 5

$\mathcal{L}_{DT}$	the loss of the teacher network training
$\mathcal{L}_{SS}$	the loss of the student network training
λ_1	the level-1 feature
λ_2	the level-2 feature
λ_3	the level-3 feature
α_1	the learning rate of the teacher network training
α_2	the learning rate of the student network training
β_1	the smoothing factor
β_2	the smoothing factor
$v_{m,n}$	the initial indicator of the n-th parameter in the m-th convolution layer
$\omega_{m,n}$	the value of the n-th parameter in the m-th convolution layer
$\mu_{m,n}$	the growing indicator of the n-th parameter in the m-th convolution layer
$p_{m,n}$	the first order momentum of the parameter at current iteration
$q_{m,n}$	the second order momentum of the parameter at current iteration
$p'_{m,n}$	the first order momentum of the parameter at last iteration
$q'_{m,n}$	the second order momentum of the parameter at last iteration
τ_m	the initial pruning threshold for the m-th convolution layer
τ_m^g	the growing threshold for the m-th convolution layer
τ_m^p	the pruning threshold for the m-th convolution layer
K_1	the maximum iteration of the teacher network training
K_2	the maximum iteration of the student network training
θ_{DT}	the parameter of DT network
θ_{SS}	the parameter of SS network
θ_{FCB}	the parameter of FCB block

N_1	the batch size of the teacher network training
N_2	the batch size of the student network training
I_t	the target domain test data sample
ξ_t	the similarity score for of target domain test sample
δ	the threshold for inference
A_m	the set of active parameter indexes for the m-th convolution layer
B_m	the set of inactive parameter indexes for the m-th convolution layer
P_m	the set of active parameter indexes of the m-th convolution layer for pruning
G_m	the set of inactive parameter indexes of the m-th convolution layer for growing
D_{src}	the source domain training data
D_{tgt}	the target domain training data
i	the index of data sample
x_i	the image of the i-th source domain data sample
y_i	the label of the i-th source domain data sample
z_i	the image of the i-th target domain data sample
d_i	the estimated pixel map of the ith data sample
k	the index of training iteration
n	the index of parameter in convolution layer
l_m	the number of parameters in the m-th convolution layer
m	the index of the convolution layer
f	the teacher network feature
f'	the student network feature
f_i^1	the teacher network level-1 feature of the i-th data sample
f_i^2	the teacher network level-2 feature of the i-th data sample
f_i^3	the teacher network level-3 feature of the i-th data sample
$f_i'^1$	the student network level-1 feature of the i-th data sample
$f_i'^2$	the student network level-2 feature of the i-th data sample
$f_i'^3$	the student network level-3 feature of the i-th data sample
$s\%$	the desired density of the SS network
$r\%$	the initial regrowth rate
T	the regrowth period

Abbreviations

ACER	Average Classification Error Rate
AIU	Adaptive Inner-Update
AMT	Asymmetric Modality Translation
AP	Attack Potential
APCER	Attack Presentation Classification Error Rate
AUC	Area Under Curve
BCE	Binary Cross-Entropy
BPCER	Bonafide Presentation Classification Error Rate
BSIF	Binarized Statistical Image Features
CDC	Central Difference Convolution
CI	Cross-Illumination
CNN	Convolutional Neural Network
COO	Coordinate
CS	Client-Specific
CUDA	Compute Unified Device Architecture
D	depth modality
DC	Discrimination
DL	deep learning
DT	Discriminative Teacher
DTN	Deep Tree Network
EB	Eyes-Bent

EER	Equal Error Rate
ENMB	Eyes-Nose-Mouth-Bent
ENMS	Eyes-Nose-Mouth-Still
ENB	Eyes-Nose-Bent
ENS	Eyes-Nose-Still
ES	Eyes-Still
FAR	False Acceptance Rate
FDR	False Detection Rate
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FRR	False Rejection Rate
FT	Fusion Training
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
HSV	hue, saturation, value
I	near infrared modality
I2I	image-to-image
IN	Illumination Normalization
IN-VIS	VIS image processed by IN module
IMQ	Image Quality
K	thousand
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LOO	leave-one-out
LTO	leave-three-out
MLS	meta loss function search
MMD	maximum mean discrepancy
NAS	network architecture search
NIR	near infrared

NN	neural network
OCDA	one-class domain adaptation
OCSVM	one-class support vector machine
PA	presentation attack
PAD	presentation attack detection
PLGF	Pattern of Local Gravitational Force
QDA	Quadratic Discriminant Analysis
rPPG	remote Photoplethysmography
ReLU	Rectified Linear Unit
RGB	red, green, blue
ROC	receiver operating characteristic
SEF	Squeeze-and-Excitation fusion
SLR	Sparse Logistic Regression
SOTA	state-of-the-art
SS	Stubborn Student
SVM	support vector machine
SWIR	short wave infrared
T	thermal modality
TDR	True Detection Rate
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
UV	ultraviolet
V	visible light modality
VIS	visible light
VLAD	vector of locally aggregated descriptor

Summary

Face recognition as a convenient approach for automatic identity verification has become increasingly prevailing in recent years. The presentation attack (PA) is a serious threat hindering the application of face recognition systems in security-critical scenarios. Face presentation attack detection (PAD) is an essential anti-spoofing measure to enhance the security of face recognition systems by discriminating presentation attacks from bona fide attempts. Existing methods have achieved good performance in intra-domain testing, where the testing data is from the same distribution as training data. However, when testing the face PAD models in a new target domain, the performance will degrade severely since the testing data could be from unseen distributions which are different from the training data.

In this thesis, we explore the cross-domain problems in face PAD and introduce several methods to apply to different application scenarios. In consideration of the intrinsic difference between genuine face and attack samples, such as the feasibility and the expense of data collection in practical scenarios, our methods are devised with more attention to genuine face samples. Considering that the attackers may launch presentation attacks with novel spoofing mediums, we study the unseen attack problem in face PAD in the first work and propose method based on deep metric learning. We learn a discriminative feature space with a hypersphere loss which forces the genuine face samples to maintain intra-class compactness and ensure inter-class separation from the attack samples. Since the decision-making is directly conducted on the learned feature space, there is no need for additional classifiers to be trained. Beyond the threats of unseen attacks, the changes in illumination conditions and camera sensors will also degrade the reliability of the face PAD systems. In the second work, we tackle the generalization problems in

face PAD and propose a bi-modality method that better generalizes to unseen attack and illumination variations. We establish the connection between face images of different modalities via asymmetric modality translation. The discrepancy of modality translation between genuine faces and attack samples is used as a compelling clue for discriminating various spoofing faces from genuine faces. Domain adaptation is a typical approach to improving the cross-domain performance of face PAD with the help of target domain data. However, it has always been a non-trivial challenge to collect sufficient data samples in the target domain, especially for attack samples. In the third work, we improve the cross-domain performance of the face PAD by only using a few genuine face samples collected in the target domain. We propose a method by introducing teacher-student learning to address the one-class domain adaptation problem in face PAD. The similarity score between the representations of the teacher and student networks is used to distinguish attacks from genuine ones.

To verify the effectiveness of the proposed methods, we devise protocols and conduct extensive experiments on multiple datasets. The experimental results show that our methods outperform prior methods.

Chapter 1

Introduction

1.1 Background of Face Presentation Attack Detection

Face recognition has become one of the most predominant approaches for automatic identity verification due to its high accuracy and convenience [1]. The noncontact and silent sensing process makes it suitable for diverse application scenarios, from attendance registration and mobile device unlocks to security-critical ones such as door access control, border control, and cashless payment.

With the wide application of face recognition techniques in various scenarios, the security and trustworthy problems of face recognition systems have been widely concerned in academia and industry. Despite the ease of use, face image has higher disclosure risks compared to fingerprints and iris images, especially with the widespread use of social media. The presentation attack, also known as the spoofing attack, is a physical-layer attack that works on the sensing process of face recognition systems, which aims to spoof the systems and be verified as the attempts of genuine users. As the name implies, it is conducted by presenting facial forgery (printed photos, images or videos on digital displays, high-fidelity masks, expensive wax figures, etc.) of genuine users to the camera sensor of face recognition systems at the sensing process. On account of the easy deployment, presentation attack (PA) is a serious threat hindering the application of face recognition systems. In addition to presentation attacks, face recognition systems are

also under the threat of some cyber-layer attacks in forms of intercepting and replacing the captured face images, replacing the probe sample features, overriding the signal processing, comparator or the decision module and etc [2]. Unlike presentation attacks targeted on the sensors, the cyber-layer attacks are related to the integrity of the system and communication networks. In this thesis, we only focus on presentation attacks.

Face presentation attack detection (PAD) is an essential anti-spoofing measure to enhance the security and reliability of face recognition systems by discriminating presentation attacks from bona fide attempts [2]. The task of face PAD is presented "face" perception and discrimination. In the past decade, various face PAD methods have emerged in the literature, ranging from the early traditional methods based on handcrafted features to recent deep learning methods with convolutional neural networks (CNN). Existing face PAD methods have achieved good performance in intra-domain testing, where the testing data is from the same distribution as training data. However, when testing the face PAD models in a new target domain, the performance will degrade severely since the testing data is from unseen distributions which are different from the training data. This problem is also known as the distribution shift or domain shift problem, which originates from various factors such as the change of capturing devices, mediums of attacks, and illumination [3, 4].

1.2 Major Contributions

Since the domain shift seriously affects the reliability of face PAD systems, domain generalization and adaptation problems have become hot spots of face PAD research. In this thesis, we deal with the cross-domain problems in face PAD and propose three methods. The three methods are proposed for different application scenarios and tackle the face PAD problems with different techniques. The first and third methods apply to single visible light modality scenario. Specifically, the first method addresses the unseen attack problem based on deep metric learning and the third method addresses the domain adaptation problem based on one-class knowledge distillation. The second method tackles the generalization problem under a bi-modality scenario with the asymmetric modality translation. In consideration of the intrinsic difference between genuine face and attack samples, such as the

feasibility and the expense of data collection in practical scenarios, our methods are devised with more attention to genuine face samples. Our main contributions are stated as follows:

- **Unseen Face PAD with Hypersphere Loss**:
  The first work explores the unseen attack problem in face PAD. Unlike prior works, we propose a face PAD method based on deep metric learning, which detects attacks directly on the learned feature space without needing additional classifiers to be trained. The mapping from the image space to the target feature space is implemented with a CNN-based feature extractor. To force the genuine face samples to maintain intra-class compactness and ensure inter-class separation from the attack samples, we devise a hypersphere loss function to direct the optimization of the feature extractor. To verify the effectiveness of the proposed method, we did extensive experiments on multiple prevailing datasets. The experimental results demonstrate that our method effectively detects the unseen attack and outperforms prior methods.

- **Asymmetric Modality Translation for Face PAD**:
  In the second work, we tackle the generalization problems in face PAD under bi-modality scenarios and propose a method that generalizes better to unseen attack and illumination variations. Different from prior methods, our method explicitly establishes the connection between different modality images via asymmetric modality translation. In this method, we devise an asymmetric modality translator which successfully translates genuine face images from the source modality to the target modality while fails for attacks. The discrepancy of modality translation between genuine face and attack samples is utilized as an effective clue for discriminating various spoofing faces from genuine faces. To achieve the expected functionalities, we devise an asymmetric modality translation loss to supervise the training of the translator at both latent and pixel-level. Besides, we design an illumination normalization module based on the pattern of local gravitational force (PLGF) descriptor to reduce the effect of illumination variations on sensitive modalities. We conduct extensive experiments to verify the effectiveness of our proposed method. The results show that our method applies to different modality settings and achieves state-of-the-art performance with grand-test, cross-illumination, and unseen-attack evaluation protocols.

- **One-Class Knowledge Distillation for Face PAD**:
  In the third work, we introduce a teacher-student framework to address the one-class domain adaptation problem in face PAD. In this method, a teacher network is trained with source domain samples to provide discriminative feature representations for face PAD. Student networks are trained to mimic the teacher network and learn similar representations for genuine face samples of the target domain. In the test phase, the similarity score between the representations of the teacher and student networks is used to distinguish attacks from genuine ones. To relieve the pressure about the expansion of the model size, a sparse learning strategy is adopted during the training of student networks. To evaluate the face PAD models under one-class domain adaptation settings, we devise two new protocols and conduct extensive experiments. The experimental results show that our method outperforms baseline methods with one-class domain adaptation and also unsupervised domain adaptation.

1.3 Organization of the Thesis

Chapter 1 introduces the background and challenge of the face presentation attack detection task, the major contribution of our work, and the organization of the thesis.

Chapter 2 reviews the evolution of face presentation attack detection methods, ranging from the conventional methods based on handcrafted features to recent deep learning methods that are more relevant to our work.

Chapter 3 introduces the work that tackles the unseen attack problems in face PAD with a method based on deep metric learning.

Chapter 4 introduces the work that tackles the generalization problems in face PAD with a method based on asymmetric modality translation.

Chapter 5 introduces the work that tackles the one-class domain adaptation problem in face PAD with a method based on teacher-student learning.

Chapter 6 summarizes our works included in the thesis and discusses the directions for future works.

Chapter 2

Literature Review

Face presentation attack detection (PAD) is a task of presented "face" perception and discrimination. A diversity of sensor devices are used for information acquisition in the literature. In addition to the most commonly used visible light (VIS) camera, near-infrared (NIR) [5–9], depth [6–9], thermal [8, 9], short-wave infrared (SWIR) [9], ultraviolet (UV) [10], light-field [11, 12] and dual pixel [13] cameras have also been used for face anti-spoofing. According to the sensor devices used in the test phase, face PAD methods can be broadly categorized into single-modality and multi-modality methods. Due to the wide application of VIS cameras on multifarious daily-used electronic devices, single VIS-modality methods naturally became the mainstream of face PAD research. Although single VIS-modality methods have been extensively studied for many years, the cross-domain problem of face PAD is still an open issue. Recently, multi-modality methods [6, 8, 9] have achieved promising performance with the help of richer information gathered from different modalities [5–9].

In the following content of this chapter, our review focuses on the methods based on single VIS-modality and multi-modality, which are most relevant to our work.

2.1 Single VIS-Modality Face PAD

In the past decade, various VIS-modality face PAD methods have emerged in the literature, ranging from the conventional methods based on handcrafted features to recent deep learning methods.

2.1.1 Conventional Methods

At the early stage, face PAD models are constructed with handcrafted features and conventional classifiers like most other computer vision tasks.

2.1.1.1 Feature Representations

In the literature, a variety of handcrafted features have been devised to represent the data samples in discriminative feature spaces that are easier for classifiers to distinguish between genuine and spoof face images. The handcrafted features are designed by experts with specific experience and domain knowledge. Motivated by the observation that the genuine face images contain more high-frequency details than recaptured images, some earlier works [14–16] extract handcrafted features by analyzing the Fourier spectra. Since the spoofing face images usually have lower image quality and contain more texture defects, there are some features devised on the basis of image quality analysis [17–19] and micro-texture analysis [20–23].

Beyond the face PAD methods based on single frame images, there are some works extracting feature representations from image sequences. Facial motions such as eyeblink [24] and mouth movement [25] have been used as cues for differentiating spoof artifacts from genuine faces. Earlier methods extract motion information from the image sequences with the help of optical flow analysis [25–27]. There are also some works that extract dynamic features [28–32] with spatial-temporal descriptors extended from conventional descriptors in texture and spectral analysis. Besides, heartbeat signals such as remote Photoplethysmography (rPPG) extracted from face image sequences have been used to construct discriminative features for face PAD in recent works [33–36].

2.1.1.2 Classifiers

In addition to the feature representation, the classifier is also an essential component of conventional face PAD methods. In the literature, the classifiers can be broadly categorized as binary and one-class classifiers. Binary classifiers are constructed with features of both genuine face and attack samples. Support Vector Machine (SVM) [37] is one of the most common classifiers in machine learning, which is also widely used in face PAD [19–23, 26–28, 31–36]. Beyond SVM, some classifiers based on Sparse Logistic Regression (SLR) [38], Linear Discriminant Analysis (LDA) [39], and Quadratic Discriminant Analysis (QDA) [39], AdaBoost [40] have also been used in conventional face PAD methods [15–18, 24, 25]. Unlike binary classifiers, one-class classifiers are built with features of genuine face samples only in scenarios where the attack samples are unreliable or even unavailable. In the literature, one-class classifiers such as one-class support vector machine (OCSVM) [41] and Gaussian mixture model (GMM) [42] have been used to detect presentation attacks under the unseen attack setting [43–45].

Although handcrafted features have strong interpretability, their representational ability is limited. The designs of these features are usually based on prior knowledge about specific types of attacks. Rare of them can take care of various types of attacks, especially when the presupposed cue does not exist. Moreover, the discrimination ability of conventional classifiers is limited compared to recent neural network based discriminators.

2.1.2 Deep Learning Methods

With the great success of deep learning techniques in the field of representation learning and various computer vision tasks, convolutional neural networks have been used for face PAD and demonstrated significant advantages and great potential. At the earlier stage, CNNs designed for general computer vision tasks were used as the feature extractor to replace the conventional handcrafted feature descriptors. Yang *et al.*'s work [46] is the first attempt at introducing deep learning techniques in face PAD task. Instead of using handcrafted features, they utilize genuine face and attack images to learn feature representations with a feature extractor based on VGG-Net [47] and construct SVM based classifiers for face PAD.

After which, an increasing number of deep learning based face PAD methods have emerged along with the rapid development of deep learning techniques. The good performance of deep learning based methods relies on the design of neural network architectures and the optimization objectives, as well as the diversity of training data and the quality of annotations.

2.1.2.1 Network Architectures

Yu *et al.* [48] propose a network architecture based on central difference convolution (CDC) layers for fine-grained feature learning. Network architecture search (NAS) techniques have been used to automatically search the optimal parameters of network architectures for higher accuracy [48, 49] and efficiency [50]. Despite existing data-driven face PAD methods performing well on intra-domain testing data, the excellent performance is limited by the diversity of the source domain training data, and seldom of them can generalize well on the target domain due to the domain shift problem [51] caused by complex factors. Beyond methods that learn feature representations from a single image frame, there is also a branch of face PAD methods devised with spatial-temporal feature learning [49, 52–58].

2.1.2.2 Auxiliary Tasks

Since CNN-based face PAD models naively optimized with binary classification loss usually suffer from overfitting problems, auxiliary tasks have been used to assist the training of face PAD models. Motivated by the evidence that genuine and spoofing faces have different geometry shapes, depth regression [48, 59–63] is widely used as an auxiliary task in recent deep learning based face PAD methods. Besides, the reflection pattern [61, 62], texture map [64], binary mask [65], and rPPG signal [60] regression have also been used to assist the training of face PAD models. To avoid the inconvenience of fine-grained label generation, George *et al.* [66] propose to train a CNN-based face PAD model using pixel maps with 0/1 value. Motivated by the observation that spoofing face images contain intrinsic defects, some methods construct face PAD models with the help of noise [67] and spoofing traces modeling [68, 69]. Similar to conventional methods, there are also some deep learning based face PAD designed from the perspective of intrinsic image decomposition [70, 71] and reflection analysis [72]. Since the spoofing attacks are launched by presenting

spoofing artifacts to the camera sensors, the captured face images usually contain the contours of spoof mediums. Zhu *et al.* [73] formulate face PAD as a task to detect spoofing medium contours directly. Cai *et al.* [74] formulate face PAD task in a global-to-local manner with deep reinforcement learning.

2.1.2.3 Frequency Domain Analysis

Frequency domain analysis have been widely used in image forensics [75, 76]. Some recent face PAD works [77–79] transform the face images from the spatial domain to the frequency domain to extract features that are more effective to capture subtle spoofing artifacts and robust to the capturing environment variations. Chen *et al.* [77] propose a two-stream face PAD model which extracts high and low frequency information from the face images and fuses the information using a cross-frequency spatial attention (CFSA) module for better generalization performance. To alleviate the performance loss of face PAD caused by the variance from camera sensors, Chen *et al.* [78] propose a framework which learns camera-invariant features for face PAD by performing feature decomposition and re-composition in the frequency domain. Considering the filters used in prior frequency-based face PAD methods are fixed-weighted and sub-optimal, Fang *et al.* [79] propose a face PAD framework which introduces learnable filters to learn more robust features in the frequency domain.

2.1.2.4 Domain Generalization Techniques

The performances of plain data-driven methods are limited by the diversity of training data [3]. These methods can not generalize well under the domain shift caused by the variations of illumination conditions and capture devices. Since the domain shift problem severely interferes with the reliability of face PAD models, domain generalization and adaptation techniques have been applied in the face PAD task to improve the target domain performance in recent years.

Domain generalization based methods assume that the target domain data is unavailable for the model training and aim to develop a generalized model by utilizing data samples from multiple source domains. Li *et al.* [53] design a generalization loss that guides the neural network to learn generalized feature representation by

manipulating the feature distribution distances of different data domains. Disentangled representation learning techniques [80, 81] have been used to learn domain-independent features for generalized face PAD models. Wang *et al.* [80] propose a framework with a disentangled representation learning module and a multi-domain learning module to force the feature representations of the face PAD model to more subject-independent and domain-independent. Wang *et al.* [81] argue that using simple global pooling makes the representations of face PAD models lose local feature discriminability. Therefore, they propose a framework based on a modified vector of locally aggregated descriptors (VLAD) to learn better feature representations. Considering the biased distribution of different data domains, Liu *et al.* [82] propose a framework that iteratively reweights the relative importance between samples with two different reweighting modules. Besides, there are several meta-learning frameworks [83–87] have been proposed to learn generalized face PAD models with specific meta tasks under the simulated train and test scenarios with domain shifts. To tackle the problem that face PAD models are easy to overfit on seen attacks, Qin *et al.* [83] propose a meta-learning framework with a Fusion Training (FT) and Adaptive Inner-Update (AIU) learning rate strategy. In [84], Shao *et al.* propose a different meta-learning framework to address the domain generalization problem. The framework simultaneously conducts meta-learning in multiple scenarios with simulated domain shifts and incorporates a regularization based on domain knowledge. Liu *et al.* [85] argue that the normalization of features greatly impacts the generalization of the learned representation. To learn more generalizable representations for face PAD, they propose a meta-learning framework with an adaptive feature normalization module for normalization method selection and dual calibration constraints to direct the model optimization. To explore better supervision for face PAD, Qin *et al.* [86] propose a bi-level optimization framework with a meta-teacher that supervises the detection network to learn rich spoofing cues. Cai *et al.* [87] propose a face PAD method which iteratively update the network for meta pattern extraction and discrimination. Most domain generalization based methods require using the domain labels during training, while manually assigning training data with different domain labels is expensive, and the partition is usually sub-optimal. To avoid this limitation, Chen *et al.* [88] propose a method without using domain labels, which iteratively divides mixture domains under a meta-learning framework.

2.1.2.5 Domain Adaptation Techniques

As complementary to the domain generalization, domain adaptation techniques have also been used for face PAD to further improve the target domain performance with some target domain training data.

Most existing works formalize the face PAD problem under the unsupervised domain adaptation setting. They aim to improve the performance of face PAD models with unlabelled target domain data. Li *et al.* [51] propose a framework to learn a more generalized classifier for face PAD by minimizing the maximum mean discrepancy (MMD) between the source and the target domain features. Wang *et al.* [89] propose a face PAD approach that uses adversarial domain adaptation techniques to improve the generalization performance. Li *et al.* [90] propose a method by introducing knowledge distillation to address the domain adaptation problem of face PAD with limited target domain training data. Jia *et al.* [91] propose a method to mitigate the distribution discrepancy between different data domains by performing marginal and conditional distribution alignment. Wang *et al.* [92] propose an approach that uses disentangled representation to improve the generalization performance of face PAD model.

Due to the difficulty and expense of data collection, collecting sufficient data samples in the target domain for the model training is unrealistic. Compared to attack samples, genuine face samples are much easier and cheaper to collect. Recently, some works aim to improve the performance of the face PAD model with only a few genuine face images of the target domain. The first attempt to use genuine face image samples to improve the target domain performance in face PAD dates back to [93], where Yang *et al.* propose a method of virtual feature generation to address the problem that generic face PAD classifier based on handcrafted features cannot generalize well to all subjects. Recently, the concept of one-class domain adaptation has been used to improve the performance of face PAD models under scenarios with domain shifts between the source and the target domains. Mohammadi *et al.* [94] propose a method that uses domain-guided pruning to adapt a pre-trained PAD network to the target dataset. Qin *et al.* [95] propose a meta-learning framework with a meta loss function search strategy to boost the performance of the face PAD model under the one-class domain adaptation setting.

2.2 Multi-Modality Face PAD

Compared to the single VIS-modality face PAD research, multi-modality face PAD is still under-explored. This is mainly due to the cost of using additional sensor devices.

Recently, the falling price of multi-modality sensors make them affordable to be equipped on the latest mobile devices. Several large-scale datasets [6–9] containing paired multi-modality data have been released in the past two years to support the study and evaluation of multi-modality face PAD methods. Zhang *et al.* [6] collect a large-scale dataset named CASIA-SURF containing data samples of VIS, NIR, and depth modalities. At the same period, George *et al.* [8] introduce another multi-modality dataset called WMCA for face PAD research. The dataset contains VIS, NIR, thermal, and depth modality data. Compared to CASIA-SURF, the WMCA dataset is more diverse in attack type, which contains seven types of attacks ranging from 2D image and video attacks to 3D masks, fake heads, and partial attacks. Recently, George *et al.* [96] extend the WMCA dataset with SWIR modality data. To support the cross-ethnic face PAD research, Liu *et al.* [7] collect a new dataset with subjects of different ethnics.

Zhang *et al.* [6] propose a method based on ResNet18 [97] with squeeze-and-excitation fusion (SEF) [98] to detect printed photo attacks in multi-modality scenarios. To further improve the performance, SEF has been applied at multiple feature levels of ResNet [97] to fuse both global and local features [99]. Besides, George *et al.* [8] propose a face PAD method with multi-modal input data. The backbone of the network is based on LightCNN [100] and the fusion of different modalities is conducted at feature level. Before long, George *et al.* [96] propose another multi-modality method named MC-DeepPixBiS. The network of MC-DeepPixBiS is constructed with the convolutional blocks proposed in DenseNet [101]. Aligned images of different modalities are concatenated along the channel axis, and the concatenation results are fed as the input to the face PAD network. The network training is supervised with both pixel-wise binary and binary labels by drawing lessons from the advances in VIS-modality face PAD method [66]. Similarly, Yu *et al.* [102] propose a method based on CDC [48] for face PAD in multi-modality scenarios. In addition to fully multi-modality methods, some face PAD methods use multi-modality data at the training phase only. The method proposed in [103]

only uses VIS images for inference at the test phase, although NIR images are used as auxiliary information to assist model training.

The majority of existing multi-modality face PAD methods are simple extensions of VIS-based ones. The intrinsic characteristics of the face PAD task and the relationships between different modalities are scarcely considered. The generalization problems of multi-modality face PAD, such as the robustness against unseen attacks and the variation of illumination conditions, are scarcely studied.

2.3 Datasets and Evaluation Metrics

In recent years, with the ceaseless emergence of new attacks and the progress of sensor technology, a lot of datasets for face PAD research have been proposed. The datasets are collected in different environments with different camera sensors and vary in sample size and attack type. To verify and compare the effectiveness of different face PAD methods, various evaluation metrics have been devised with different considerations. This section systematically introduces the datasets and evaluation metrics related to our work.

2.3.1 Single VIS-Modality Datasets

1) CASIA-FASD

The CASIA Face Anti-spoofing Database (CASIA-FASD) [104] contains 600 video clips collected with 50 subjects. It includes 150 genuine face videos and 450 attack videos. Each subject in the database has 12 corresponding video clips. The videos are collected with 3 different camera sensors, and the attack samples are in printed photos, paper masks, and digital screens.

2) IDIAP REPLAY-ATTACK

The IDIAP REPLAY-ATTACK Database [21] contains 1200 video clips of 50 subjects, including 200 genuine face videos and 1000 attack videos. With each subject, there are 24 videos collected under controlled and adverse illumination conditions. The attack samples are printed photos, digital face images, and face videos displayed on digital screens.

3) MSU-MFSD

The MSU Mobile Face Presentation Attack Database (MSU-MFSD) [19] contains 280 video clips of 35 subjects, including 70 genuine face videos and 210 attack videos. There are 8 videos with each subject. The videos are collected with two types of cameras sensors, and the attack samples are in the type of printed photos and videos replayed on iPhone and iPad.

4) NTU ROSE-YOUTU

The NTU ROSE-YOUTU Face Liveness Detection Dataset [51] contains 3350 video clips of 20 subjects, including 1000 genuine face videos and 2000 attack videos. There are 150-200 video clips for each subject with an average duration of 10 seconds. Five mobile phones with cameras of different specifications have been used for video capture. The attack samples are printed photos, face videos on digital screens, and partial paper masks.

5) OULU-NPU

OULU-NPU Database [3] contains 4950 video clips of 55 subjects, including 990 genuine face videos and 3960 attack videos. There are 90 video clips for each subject. The videos are collected by the camera sensors of 6 different types of mobile phones under 3 illumination environments. The attack samples are in the type of printed photos and face videos on digital screens.

6) SiW-M

Spoof in the Wild Database with Multiple Attack Types (SiW-M) [105] contains 660 genuine face videos and 968 attack videos in 13 attack types. In addition to the conventional printed photo and replayed video attacks, the SiW dataset also includes mask attacks, partial attacks, and makeup attacks for obfuscation and impersonation purposes.

2.3.2 Multi-Modality Datasets

1) WMCA

The WMCA dataset [8] covers genuine faces and 7 categories of attack samples. Each original video is processed into 50 images by preprocessing procedures introduced in [8]. In specific, there are 28540, 27550, 27850 samples in *train*, *dev*, *test* subset respectively. Each data sample contains images of VIS (V for short), NIR (I for short), thermal (T for short), and depth (D for short) modalities.

2) CASIA-SURF

The CASIA-SURF dataset [6, 99] contains genuine faces and 6 types of 2D attack samples of VIS (V for short), NIR (I for short), and depth (D for short) modality. The dataset has different versions, and we use the same one as [99] in our experiments for fair comparisons with previous work. For the data preprocessing scheme, original video clips are processed into cropped images of the face region by specific procedures as introduced in [6]. We use the processed data provided in the dataset directly for experiments, and images of V modality are converted to grayscale to keep consistent with WMCA and MSSPOOF. Eventually, there are about 49K, 16K, 98K samples in *train*, *dev*, *test* set respectively.

3) MSSPOOF

The Multispectral-Spoof Database (MSSPOOF) [5] contains paired images of VIS (V for short) and NIR (I for short) modalities, which are captured under various environments. It covers genuine face images, printed VIS, and NIR images. Following the data preprocessing procedures introduced in [8], we align images according to the eye-center position by using landmark annotations provided in the dataset and crop face regions. After data preprocessing, there are 946, 641, 639 pairs of samples in *train*, *dev*, *test*.

2.3.3 Evaluation Metrics

Before introducing the evaluation metrics, we firstly explain some basic concepts. In face PAD, the genuine face and attack samples are defined as negative and positive, respectively. Therefore, True Positive (TP) is defined as that the attack sample is recognized as the attack; False Positive (FP) is defined as that the attack sample is recognized as the genuine face; True Negative (TN) is defined as that the genuine face sample is recognized as the genuine face; False Negative (FN) is defined as that the genuine face sample is recognized as the attack.

1) AUC
The receiver operating characteristic (ROC) curve [106] illustrates the discrimination ability of a binary classification system by plotting the true positive rate (TPR) against the false positive rate (FPR) at different thresholds. The Area under curve (AUC) is commonly used as the metric to evaluate the comprehensive discrimination ability of face PAD models.

$$TPR = \frac{TP}{TP + FN} \tag{2.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.2}$$

2) HTER
Half Total Error Rate (HTER) is the average of False Rejection Rate (FRR) and False Acceptance Rate (FAR). FRR measures the ratio of incorrectly rejected genuine face samples and FAR measures the ratio of incorrectly accepted attack samples [2].

$$HTER = \frac{FRR + FAR}{2} \tag{2.3}$$

$$FRR = \frac{FN}{TP + FN} \tag{2.4}$$

$$FAR = \frac{FP}{FP + TN} \tag{2.5}$$

3) EER
Equal Error Rate (EER) is a point where the FRR is equal to FAR, which indicates the comprehensive performance of binary classification systems considering the trade-off between the FRR and FAR.

4) ACER
Average Classification Error Rate (ACER) is the average of Bona-fide Presentation Classification Error Rate (BPCER) and Attack Presentation Classification Error Rate (APCER).

$$ACER = \frac{BPCER + APCER}{2} \tag{2.6}$$

BPCER measures the error rate of classifying bona-fide presentations as attack presentations. The computation of BPCER is defined as:

$$BPCER = \frac{\sum_{i=1}^{N_B} RES_i}{N_B} = FPR = \frac{FP}{FP + TN} \tag{2.7}$$

where N_B is the number of bona-fide samples. RES_i is 1 when the ith sample is classified as the attack, and 0 otherwise. For a specific presentation attack species, the APCER measures the error rate of classifying attack presentations as bona-fide presentations. In the literature, the computation of the overall APCER is defined differently in the protocols of different datasets. The OULU-NPU dataset [1] defines the overall APCER as $APCER_{AP}$:

$$APCER = APCER_{AP} = \max_{PAI}\left(\frac{\sum_{i=1}^{N_{PAI}}(1 - RES_i)}{N_{PAI}}\right) \tag{2.8}$$

where N_{PAI} is the number of attack samples in the same species. Some other datasets [2] (e.g. CASIA-SURF) define the overall APCER as equal to FNR:

$$APCER = FNR = \frac{FN}{TP + FN} \tag{2.9}$$

In our experiments, we strictly followed the protocols of each datasets.

5) TDR@FDR=1%
The True Detection Rate (TDR) and False Detection Rate (FDR) are computed as TPR and FPR, respectively, if the attack sample is defined as positive. The TDR@FDR=1% measures the sensitivity of face PAD models at the tolerance of that the models incorrectly classify 1% genuine face samples as attacks.

$$TDR = \frac{TP}{TP + FN} \tag{2.10}$$

$$FDR = \frac{FP}{FP + TN} \tag{2.11}$$

[1] http://jultika.oulu.fi/files/nbnfi-fe2019091228006.pdf
[2] https://sites.google.com/qq.com/face-anti-spoofing/evaluation

Chapter 3

Unseen Face PAD with Hypersphere Loss

3.1 Introduction

Most previous works formulate face presentation attack detection (PAD) as a close-set binary classification problem in the ideal laboratory scenario. They assume that sufficient data samples are available for model training, and all types of attacks encountered in the test phase are seen in the training phase. However, the practical scenario of face PAD is more complex since the attackers may employ novel attacks that are never seen by the face PAD model before. Hence, there is a necessity to develop face PAD models that are effective against unseen types of attacks.

Face PAD problem has been formulated under the unseen attack setting in recent work [43]. Arashloo *et al.* [43] construct a set of face PAD models by combining different handcrafted features with conventional one-class or binary classifiers to evaluate the effectiveness of the models against unseen types of attacks. Soon afterward, Nikisins *et al.* [44] propose a face PAD method based on image quality features [18, 19] and one-class classifiers. Similarly, color textures [3] have been used as feature representation to construct face PAD models with different one-class classifiers in [45]. Fatemifar *et al.* [107] propose to develop client-specific face PAD models with pre-trained CNNs as the feature extractors. The research on the face PAD problem under the unseen attack setting is still in the earlier stage. The experiments of most prior works are conducted on conventional face PAD datasets

with limited types of attack samples. To address the limitation of insufficient attack diversity of existing datasets, Liu *et al.* [105] collect a dataset containing 13 types of attacks to support the evaluation of face PAD methods under a more realistic experimental setting. A deep tree learning based face PAD method [105] has been proposed to hierarchically routes the image samples into different branches for final classification.

In this work, we aim to address the unseen attack problem in face PAD. Our main contributions in this work [1] can be summarized as below:

- We propose a deep metric learning based method for face PAD under the unseen attack scenario. Our method end-to-end trains a CNN-based model, which transforms face images to feature presentations and detects attacks directly on the learned feature space with no need for additional conventional classifiers to be trained.
- We devise a hypersphere loss function to direct the optimization of the CNN-based model.
- We conduct extensive experiments to evaluate the performance of our proposed method. The results show that our method generally outperforms prior methods.

3.2 Methodology

Before introducing the proposed method, this section firstly elaborates on the problem we are working on. We aim to address the unseen attack problem in face PAD task. Specifically, the task is to develop a generalized face PAD model by utilizing genuine face and available attack samples. Considering that the type of attacks encountered in the testing phase may not be the same as the samples seen in the model training phase, the face PAD model is expected to be generalized and effective against unseen types of face presentation attacks.

Since the available information about genuine faces and attacks is not equivalent, we believe it is not appropriate to treat the face PAD as a conventional binary classification problem. Firstly, compared with genuine face samples, attacks are more

[1] The work in this chapter has been published in [108]

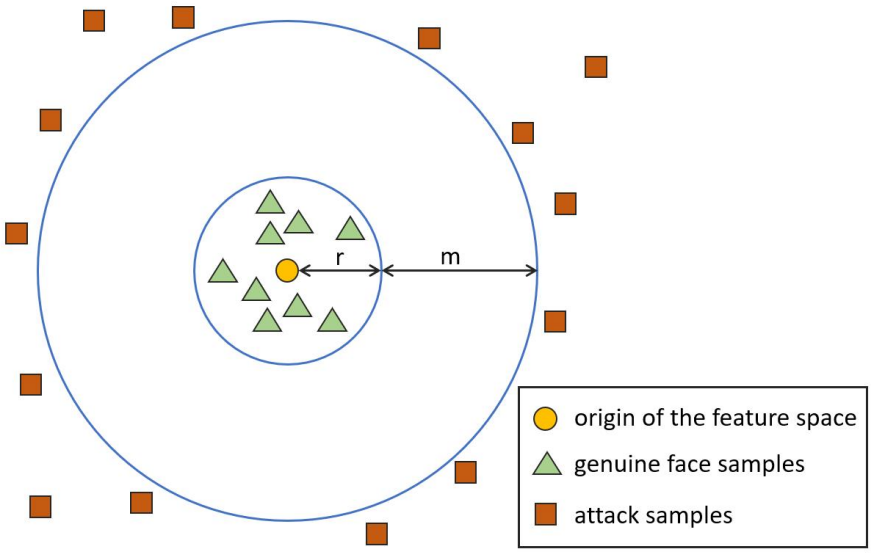
variable. The variation of genuine face samples is mainly caused by the changes in illumination conditions and camera sensors. In addition to the aforementioned factors, attack samples are diverse in spoof mediums. The difference results in the situation that genuine face samples should naturally form a cluster keeping intra-class compactness which is not applicable to attack samples. Secondly, the task is under the unseen attack setting due to the uncontrollable characteristics of attacks encountered in the testing phase. Attack samples for model training are not as reliable as genuine face samples and could be used as the simulation of real attacks only.

We believe that training the face PAD models with conventional binary classification loss functions will result in the over-fitting problem on attack samples seen at the training phase. Therefore, we propose a face PAD method based on deep metric learning with a hypersphere loss function. Different from prior methods based on one-class and binary classifiers, our method learns a CNN-based feature extractor to embed the face images into a latent feature space and detects attacks on the learned feature space directly.

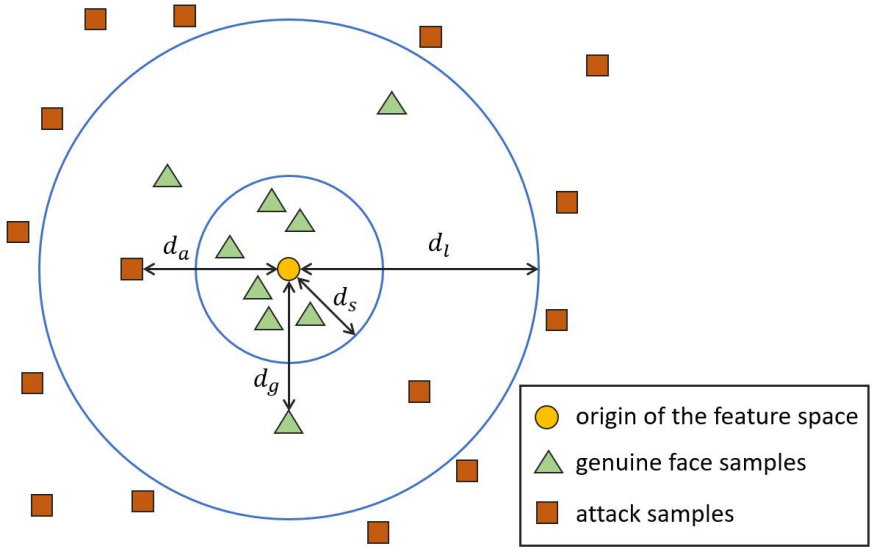
3.2.1 Hypersphere Loss Function

With the aforementioned analysis, we start to introduce the hypersphere loss function by describing the target feature space.

As is shown in Figure 3.1 (A), genuine face samples should be distributed near the origin and converge to a small hypersphere of radius r to maintain intra-class compactness. On the contrary, attack samples are expected to be away from the smaller hypersphere by a predefined margin m to ensure inter-class separation between the genuine face and attack samples. In our design, the loss value for genuine face and attack samples are computed in different ways. As shown in Figure 3.1 (B), we represent loss by using the distance between the data sample and its expected position. The square of L_2 norm is used for distance calculation. If there are N_g genuine face samples and N_a attack samples, the loss of genuine face samples $\mathcal{L}_g$ and attack samples $\mathcal{L}_a$ are represented as Eq.(3.1) and Eq.(3.2), respectively.



(A) Target Feature Space



(B) Distances in the Hypersphere Loss

FIGURE 3.1: The figure illustrates the target feature space and the proposed hypersphere loss. As shown in figure (A), the genuine face samples are expected to converge to a small hypersphere of radius r while the attack samples should be separated from the smaller hypersphere by a margin m. The figure (B) illustrates the distances in the proposed hypersphere loss.

$$\mathcal{L}_g = \sum_{i=1}^{N_g} \max(d_i^g - d^s, 0) \tag{3.1}$$

$$\mathcal{L}_a = \sum_{i=1}^{N_a} \max(d^l - d_i^a, 0) \tag{3.2}$$

where $d_i^g = \|f_i^g\|_2^2$, $d_i^a = \|f_i^a\|_2^2$, $d^s = r^2$, $d^l = (r+m)^2$.

Since $\mathcal{L}_g$ and $\mathcal{L}_a$ are positively relevant to the number of data samples, the learned feature space will be biased if the amounts of the genuine face and attack samples are not comparable. As represented in Eq.(3.3), we use $\frac{1}{N_g}$ and $\frac{1}{N_a}$ to eliminate the dependence on the amount of data samples. In addition, λ_n and λ_a are hyperparameters to control the weights of loss contributed by samples of different types.

$$\mathcal{L}_h = \frac{\lambda_g}{N_g}\mathcal{L}_g + \frac{\lambda_a}{N_a}\mathcal{L}_a \tag{3.3}$$

By substituting Eq. (3.1) and Eq. (3.2) into Eq. (3.3), we get the final hypersphere loss function below.

$$\mathcal{L}_h = \frac{\lambda_g}{N_g}\sum_{i=1}^{N_g} \max(\|f_i^g\|_2^2 - r^2, 0) + \frac{\lambda_a}{N_a}\sum_{i=1}^{N_a} \max((r+m)^2 - \|f_i^a\|_2^2, 0) \tag{3.4}$$

Considering the inherent characteristics of the face PAD task, we calculate the hypersphere loss based on the distance from the origin of the feature space rather than the distance between paired data samples like typical triplet loss [109], which avoids triplet generation and hard triplet mining. Moreover, since the class center of genuine face samples is fixed at the origin, the optimization of the feature extractor is supervised by a constant learning objective. While ensuring the stable convergence of the training process, the learned feature representation is calibrated with the origin. In the testing phase, the norm of the latent feature at the learned feature space is used as the score for decision without additional classifiers to be trained.

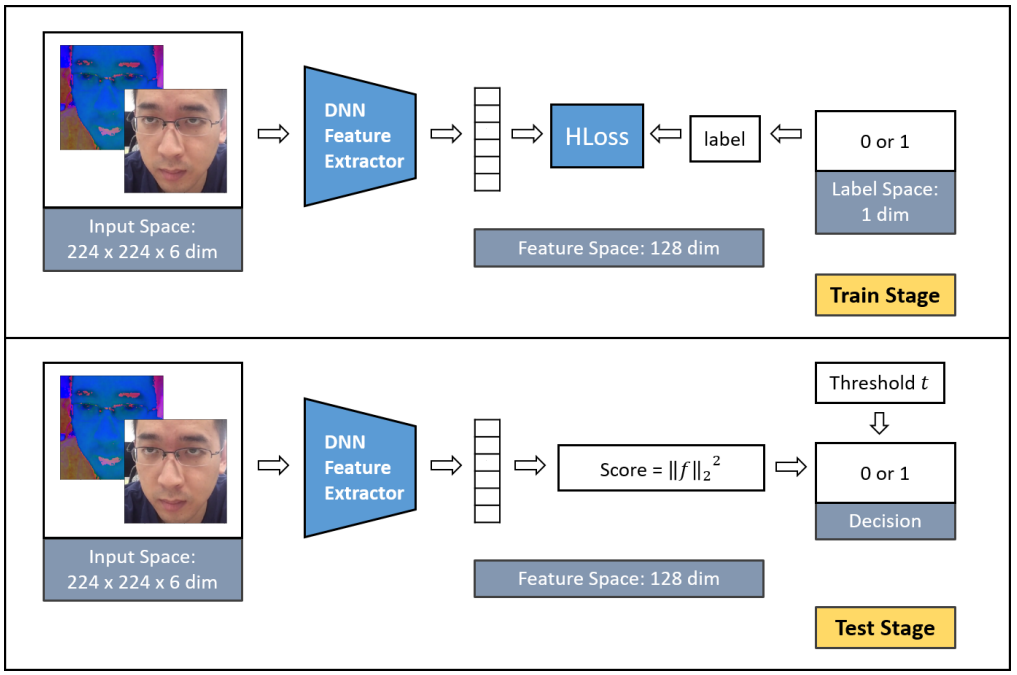


FIGURE 3.2: The figure illustrates the framework of the proposed method. The feature extractor is implemented with ResNet18, which embeds the preprocessed face image into a 128 dimensional feature space. The training of the feature extractor is directed by minimizing the hypersphere loss. At the testing phase, the squared L_2 norm of the feature vector is used as the score compared with the predetermined threshold for inference.

3.2.2 Framework of the Proposed Method

As shown in Figure 3.2, after data prepossessing such as face detection and cropping, the face images are converted into RGB and HSV colorspace. Following [105], we concatenate the images along the channel axis as the input to the feature extractor. ResNet18 [97] is used as the feature extractor to transform the data samples into a 128-dimensional feature space.

In the training phase, the optimization objective is to learn a feature extractor that can transform the face images into the desired feature space for attack detection. The training process is end-to-end supervised by minimizing the loss function represented in Eq.(3.4). During the testing phase, the preprocessed face images are fed into the trained ResNet18 model for feature extraction. The squared L_2-norm of the feature vector is compared with a predefined threshold for decision making. Since the decision is determined on feature space directly, there is no need for additional classifiers to be trained.

3.3 Experimental Setup

3.3.1 Dataset Information

To verify the effectiveness of our proposed method, we did extensive experiments on multiple datasets such as CASIA-FASD [104], IDIAP REPLAY-ATTACK [21], MSU-MFSD [19], and SiW-M [105].

3.3.2 Evaluation Protocols

Following [43, 105], the leave-one-out (LOO) protocol is adopted for unseen attack evaluation. During the training phase, one type of attack sample is excluded as an unseen attack, and all remaining types of attack samples and genuine face samples are used for model training. The performance of the face PAD model against the unseen attack samples is evaluated during the testing phase.

3.3.3 Evaluation Metrics

For a fair comparison with prior methods, we report the results using the same metrics as previous work [43, 45, 105]. The area under the receiver operating characteristic curve (AUC) is adopted as the metric for the experiments on CASIA-FASD, IDIAP REPLAY-ATTACK, and MSU-MFSD datasets. Following [105], the performance on the SiW-M dataset is evaluated by Equal Error Rate (EER) and Average Classification Error Rate (ACER).

3.3.4 Baseline Methods

1) OCSVM$_{RBF}$ + IMQ:
One-Class Support Vector Machine (OCSVM) [41] is a classical model for one-class classification problems. It is commonly used to construct classifiers with various features. OCSVM$_{RBF}$ + IMQ [43] is a face PAD method based on OCSVM with radial basis function (RBF) kernel and image quality features [17].

2) OCSVM$_{RBF}$ + BSIF:
Binarized Statistical Image Features (BSIF) [30, 110] is an image descriptor based on independent component analysis. OCSVM$_{RBF}$ + BSIF [43] is a method constructed with OCSVM and BSIF.

3) OCSVM$_{RBF}$ + LBP:
Local Binary Pattern (LBP) [111] is a texture descriptor widely used in computer vision. The LBP features computed in the different image colorspaces have also been used for face PAD [3]. Similar to other OCSVM based methods, the OCSVM$_{RBF}$ + LBP [45] is based on OCSVM and LBP [3].

4) NN + LBP:
NN + LBP [45] is a face PAD method based on a shallow neural network (NN) consists of three fully connected layers. It takes extracted LBP features [3] as the input for binary classification.

5) SVM$_{RBF}$ + LBP:
SVM$_{RBF}$ + LBP [3] is a method based on conventional support vector machine (SVM) and LBP. Different from OCSVM$_{RBF}$ + LBP [45], the SVM classifier is trained with both genuine face and attack samples.

6) Auxiliary:
Auxiliary [60] is a deep learning based face PAD method. Different from the conventional face PAD methods training neural network models with binary classification loss. The Auxiliary model is trained with fine-grained auxiliary tasks.

7) DTN:
Deep Tree Network (DTN) [105] is a face PAD method based on deep tree learning. The network is trained for face PAD with unsupervised tree learning and supervised feature learning techniques. During the testing phase, the trained model hierarchically routes the image samples into different branches for final classification.

3.3.5 Implementation Details

For all experiments, we uniformly sample 30 image frames from each video clip, and the face region is detected and cropped from the original image frames at the input to the face PAD model. The face images are transformed into both RGB

and HSV colorspaces. ResNet18 [97] is used as the feature extractor to embed the face images into a 128-dimensional feature space, and the squared $L2$-norm of the feature vector is used as the inference score. The proposed method is implemented with PyTorch. The feature extractor is end-to-end trained with a fixed learning rate of 10^{-6} and batch size of 30. We set λ_g and λ_a in Eq. (3.4) as 1 and 1, respectively. Parameters r and m which control compactness and separation are set to 2 and 10, respectively. The computation of ACER is based on threshold. In our experiments, we strictly follow the protocol of the SiW-M dataset [105]. The threshold for decision is set as 28, which is computed and fixed with the training set. The same threshold is used for evaluation in all experiments. No data augmentation technique is used to enlarge the training set in our experiments.

3.4 Results and Analysis

3.4.1 Unseen-Attack Experiments

1) Results on CASIA-FASD, IDIAP REPLAY-ATTACK, and MSU-MFSD:
Following the experimental setting proposed in [43], we first evaluate the effectiveness of our Hypersphere method against the unseen printed photo, digital image, and replay video attacks. The performance of our method is compared with several recent face PAD methods: OCSVM$_{RBF}$ + IMQ [43], OCSVM$_{RBF}$ + BSIF [43], OCSVM$_{RBF}$ + LBP [45], NN + LBP [45], SVM$_{RBF}$ + LBP [3], and DTN [105].

As shown in Table 3.1, our proposed method outperforms the conventional methods with different handcrafted features by a distinct margin and achieves competitive performance with the recent DTN method. Specifically, our method achieves the best AUC performance on 6/9 sub-experiments. The overall performance in terms of mean and standard deviation is also better than compared methods.

2) Results on SiW-M:
As stated in [105], the variety of attack types is limited in CASIA-FASD, IDIAP REPLAY-ATTACK, and MSU-MFSD datasets, where only 2D attacks are included. To verify the effectiveness of the proposed method under a more practical and challenging experimental setting, we also test our method on the recently

TABLE 3.1: Unseen-Attack Experimental Results AUC(%) on CASIA-FASD, IDIAP REPLAY-ATTACK, and MSU-MFSD Datasets

Method	CASIA-FASD			IDIAP REPLAY-ATTACK			MSU-MFSD			Overall
	Video	Cut Photo	Warped Photo	Video	Digital Photo	Printed Photo	Printed Photo	HR Video	Mobile Video	
$OCSVM_{RBF}$ + IMQ [43]	68.9	62.0	74.8	98.2	90.8	53.2	63.9	63.0	76.4	72.8±14.5
$OCSVM_{RBF}$ + BSIF [43]	70.7	60.7	95.9	84.0	88.1	73.7	64.8	87.4	74.7	78.7±11.7
$OCSVM_{RBF}$ + LBP [45]	91.2	82.3	65.6	91.6	85.0	87.2	71.5	96.9	93.6	85.0±10.4
NN + LBP [45]	**94.2**	88.4	79.9	99.8	95.2	78.9	50.6	99.9	93.5	86.7±15.6
SVM_{RBF} + LBP [3]	91.5	91.7	84.5	99.1	98.2	87.3	47.7	99.5	97.6	88.6±16.3
DTN [105]	90.0	97.3	97.5	99.9	99.9	**99.6**	**81.6**	99.9	97.5	95.9±6.2
Ours [108]	92.7	**97.5**	**98.0**	**99.9**	**99.9**	98.7	80.2	**99.9**	**99.3**	**96.2±6.1**

TABLE 3.2: Unseen-Attack Experimental Results on SiW-M

Metric (%)	Method	Attack Type													Overall
		Replay	Print	Mask Attacks					Makeup Attacks			Partial Attacks			
				Half	Silicone	Trans.	Paper	Manne.	Obfusc.	Imperson.	Cosmetic	Funny Eye	Paper Glasses	Partial Paper	
		1	2	3	4	5	6	7	8	9	10	11	12	13	
ACER	SVM_{RBF} + LBP [3]	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 [60]	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
	Deep Tree Network [105]	**9.8**	**6.0**	15.0	18.7	36.0	**4.5**	**7.7**	48.1	11.4	14.2	19.3	**19.8**	8.5	16.8±11.1
	Ours [108]	11.6	13.0	**13.9**	23.5	**12.0**	8.6	11.2	**40.3**	**10.9**	13.1	**17.9**	20.8	8.2	**15.8±8.6**
EER	SVM_{RBF} + LBP [3]	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 [60]	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
	Deep Tree Network [105]	**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
	Ours [108]	13.2	14.0	18.1	24.0	**12.4**	**3.1**	**6.2**	**34.8**	**3.1**	16.3	21.4	21.7	9.3	**15.2±9.0**

released SiW-M dataset with 13 types of attack samples. Following [105], we report the performance with ACER and EER as evaluation metrics and compare our method with SVM_{RBF} + LBP [3], Auxiliary [60], and DTN [105] methods. The experimental results are shown in Table 3.2.

Our proposed method achieves the best overall performance in terms of both ACER and EER. Compared with DTN, our method has around 1% performance improvement in terms of the average ACER and EER. Moreover, the more minor standard deviations show that our method is more generalized to different types of attacks. As shown in Table 3.2, the superiority of our method is more evident in experiments that prior methods cannot generalize well.

For better analysis, we also visualize the performance of our method over different thresholds. Figure 3.3 plots the variation of the ACER, APCER, and BPCER performance over different thresholds in a range of 0 to 200. With the increase of the threshold, the APCER rises while the BPCER declines. The average APCER meets the average BPCER where the threshold is around 65 and the optimal threshold for average ACER is about 28. In addition to the overall performance, we also plot the ACER at different thresholds for each sub-experiment as shown in Figure 3.4. There are a total of 13 curves in different colors, and each curve corresponds to one sub-experiment under the unseen attack setting. From the figure, the optimal threshold for different unseen attack experiments varies a lot. For example, the optimal threshold is around 150 for paper mask attack (attack 6), while 5 for obfuscation makeup attack (attack 8). Considering the overall performance, we set the threshold of our method at 28 for all sub-experiments.

3.4.2 Discussion

Although our method outperforms prior methods, there are some limitations.

First, the performance of our method under the unseen attack setting is not always satisfactory, especially for the experiments with unseen silicone mask attacks, obfuscation makeup attacks, and some partial attacks. It is because these types of attack samples are more similar to the genuine face samples than the attack ones in the training set. The unsatisfactory performance is in fact the result of insufficient attack sample for model training. Given that recent generative models are able to

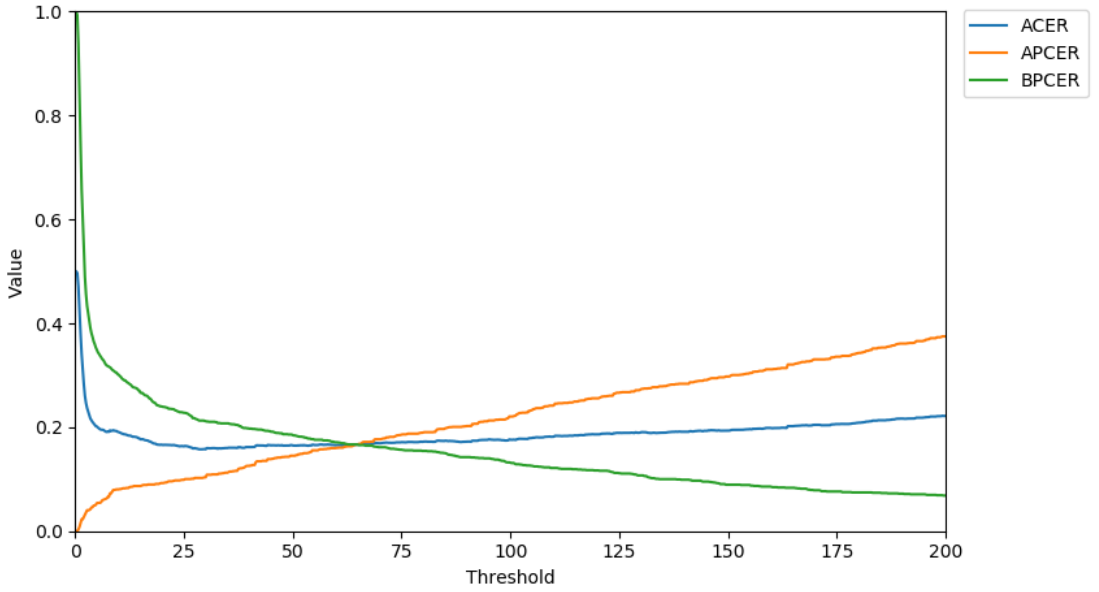


FIGURE 3.3: The figure illustrates the overall performance of our method on SiW-M dataset. The average ACER, average APCER, and average BPCER of 13 sub-experiments are plotted in blue, yellow, and green colors. The horizontal and vertical axes represent the threshold and the value of evaluation metrics, respectively.

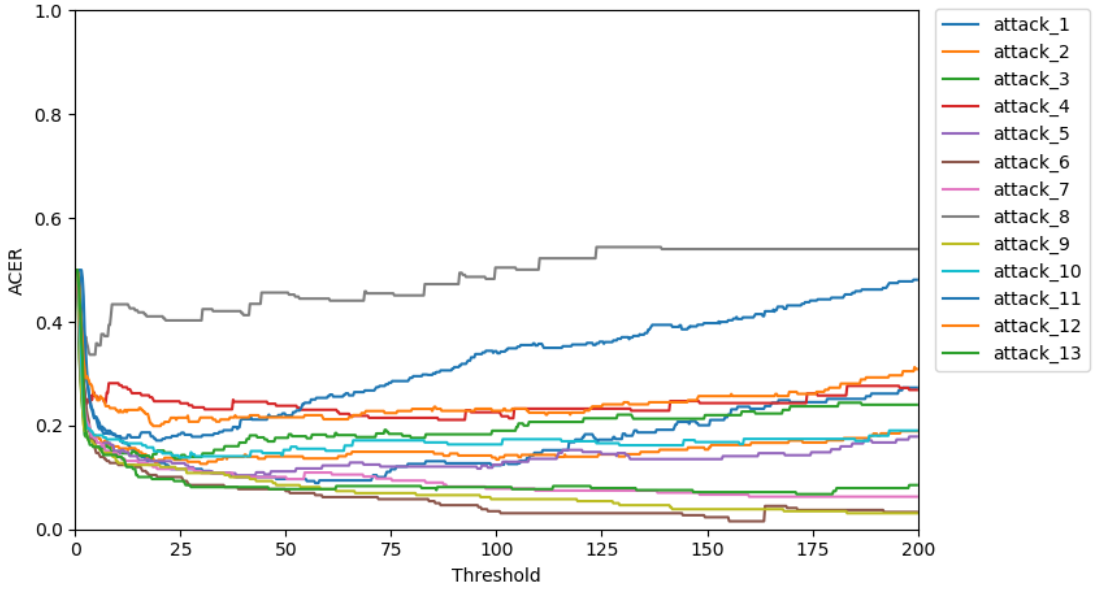


FIGURE 3.4: The figure illustrates the specific performance of our method on SiW-M dataset. The ACER of the 13 sub-experiments are plotted in different colors. The horizontal and vertical axes represent the threshold and the value of ACER, respectively.

generate realistic face images, using the data generated by these models as pseudo attack samples for model training may alleviate the issue and can be considered in future work.

Second, the optimal threshold for different unseen attack experiments varies a lot, and it is difficult to set a unified threshold that is suitable for all types of attacks. The unbounded value range of the squared L_2 norm also makes the threshold determination more empirical.

3.5 Chapter Summary

Considering the characteristics of the practical application scenarios, in this chapter, we formulate the face PAD problem under the unseen attack setting similar to recent works [43, 45, 105]. Different from prior methods, our method is based on deep metric learning. A CNN-based feature extractor is end-to-end trained with a hypersphere loss function, and the decision-making is directly based on the learned feature representations. To verify the effectiveness of our method, we did extensive experiments on multiple datasets. The performance is evaluated with different metrics and compared with several related face PAD methods. The experimental results show the superiority of our method compared with conventional ones.

However, the practical application scenarios of face PAD are more complex and challenging than the laboratory unseen attack setting. Beyond the threats of unseen attacks encountered during the testing phase, the changes in illumination conditions and camera sensors will also degrade the accuracy and reliability of the face PAD systems. How to further improve the generalization ability of face PAD is still an open issue that is widely concerned by the research community.

In Chapter 4 and Chapter 5, we introduce two methods to improve the cross-domain performance of face PAD from different perspectives.

Chapter 4

Asymmetric Modality Translation for Face PAD

4.1 Introduction

In essence, face presentation attack detection (PAD) is a task of presented "face" perception and discrimination. In the literature on face PAD research, a diversity of sensor devices have been used for information acquisition. Due to the wide application of visible light (VIS) cameras on multifarious daily-used electronic devices, VIS-based methods naturally became the mainstream of face PAD research at the early stage. Thanks to the development of sensor technology and the manufacturing process, the price of multi-modality sensors continue to drop, which makes them affordable to be equipped on up-to-date mobile devices.

Recently, multi-modality-based face PAD methods have shown promising results and become a research hot spot. Zhang *et al.* [6, 99] propose a multi-modality-based method for face PAD, which is constructed with ResNet [97] as feature extractor and squeeze-and-excitation fusion (SEF) for modality fusion [98]. At the same period, George *et al.* [8] propose a multi-channel convolutional neural network (MC-CNN) for face PAD with multi-modality sensors. In addition to the methods based on feature-level fusion, George *et al.* [96] propose the MC-DeepPixBiS method, which fuses aligned images of different modalities at the input level. Drawing lessons from the advances in VIS-based face PAD method [66], the network training is supervised with both pixel-wise binary and binary labels.

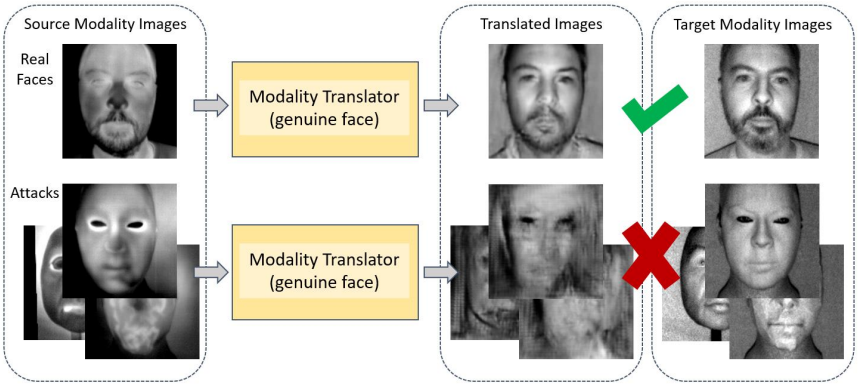


FIGURE 4.1: The figure illustrates the concept of asymmetric modality translation. We expect the modality translator can successfully translate genuine face images to the target modality while fails for attacks.

Similarly, Yu *et al.* [102] propose a face PAD method by extending CDCN [48] for multi-modality setting.

The majority of existing multi-modality-based face PAD methods are simple extensions of VIS-based ones. The intrinsic characteristics of the face PAD task and the relationships between different modalities are scarcely considered. Although existing multi-modality-based face PAD methods improve the performance to some extent, like single-modality ones, the generalization problems to unseen attacks and the variation of environment illumination still remain to be solved.

In this work, we tackle the generalization problems in face PAD under bi-modality scenarios and propose a method that better generalizes to unseen attack and illumination variations. Unlike prior works, we explicitly establish the connection between different modality images of genuine faces via asymmetric modality translation, as shown in Figure 4.1. We use it as the core to build a bi-modality face PAD framework with higher accuracy and robustness to illumination variations and unseen attacks. Our main contribution in this work [1] can be summarized below:

- We propose a method for face presentation attack detection under bi-modality scenarios with asymmetric modality translation.
- We devise an asymmetric modality translation loss to supervise the training of the translator at both latent and pixel-level and an illumination normalization module based on PLGF to reduce the effect of illumination variations on sensitive modalities.

[1] The work in this chapter has been published in [112]

- We conduct extensive experiments to verify the effectiveness of our proposed method. The results show that our method applies to different modality settings and achieves state-of-the-art performance with grand-test, cross-illumination, and unseen-attack evaluation protocols.

4.2 Methodology

Before introducing the design details, we first present the motivation of our method. For most prior multi-modality-based face PAD methods, images of different modalities are processed individually in different branches [6, 8, 99] or naively stacked together at the input level [96, 102]. The relationships between paired face images of different modalities are scarcely considered. However, we believe the relationship between different modalities could be established via cross-modality translation, and the transform functions for genuine face and attacks are different. Different from prior methods, our method aims to construct an asymmetric transform function T_G, which can successfully transform genuine face images from the source to the target modality but fails for attack ones, as illustrated in Figure 4.1. We leverage such discrepancy as an effective clue for discriminating various spoofing faces from genuine faces.

Due to the different sensing and imaging principles of different modalities, it is difficult to formulate the general transform function directly. Instead, we implement the asymmetric transform function T_G by convolution neural networks (CNN) to make it applicable to different modalities.

4.2.1 Framework of the Proposed Method

Based on the analysis above, we start to introduce the framework of our proposed method. As shown in Figure 4.2, our proposed method consists of three modules: Asymmetric Modality Translation (AMT) module, Illumination Normalization (IN) module, and Discrimination (DC) module.

The proposed method works in bi-modality scenarios, where paired images are synchronously captured using camera sensors of two different modalities. We set one modality as the source and the other as the target. After a series of preprocessing

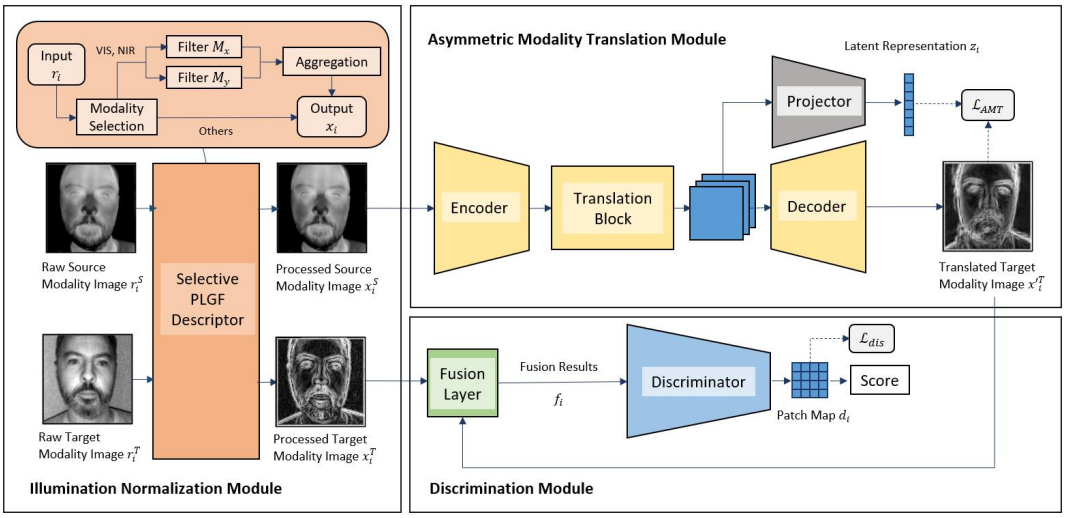


FIGURE 4.2: The figure illustrates the framework of our proposed method, which consists of three modules. The Illumination Normalization (IN) module selectively reduces the impact of illumination variations on sensitive modalities. The Asymmetric Modality Translation (AMT) module translates the source modality image to the target modality. The Discrimination (DC) module fuses the translated image with the ground-truth target modality image as the input for inference.

such as face detection-alignment-cropping-resizing, face images first pass through the IN module for illumination normalization. The source modality image is fed into the AMT module and translated into the target modality in an asymmetric way. The DC module fuses the translated image from the source modality with the ground-truth target modality image captured by the camera sensor in image space and distinguishes attack samples from genuine ones based on the output patch map. The working mechanisms and design details of each module will be introduced in the following content of this chapter.

4.2.2 Asymmetric Modality Translation

The asymmetric transform function is the core of our method. We expect the function to successfully transform genuine face images from the source to the target modality but fails for attack ones. With this objective, we seek ways to construct the expected modality translator. Image-to-image (I2I) translation methods [113–118] based on CNN have recently achieved promising results, which can automatically translate an image from the source space to the specific target space.

Leveraging the advances in the I2I translation task, we implement the asymmetric transform function with a CNN-based translator.

As illustrated in Figure 4.2, the main branch of the AMT module is a translator of AutoEncoder architecture [119] commonly used in general I2I translation tasks. The encoder network consists of 3 convolution layers with instance normalization [120], and ReLU [121] as activation function, which maps input data from the pixel space to a latent feature space. The translation block is used for modality translation at the latent space, which is implemented by several residual blocks [97]. The architecture of the decoder is symmetric to the encoder, which transforms data samples to the pixel space of the target modality. In addition to the main branch, we use an additional projector to assist in the training of the translator, which will be discarded at the inference phase. The projector consists of two convolution layers for dimension reduction.

To make the translator meet our goals, we devise an asymmetric modality translation loss to supervise the training of the translator at both pixel and latent feature spaces. In the pixel space, the outputs ${x'}_i^T$ of the decoder are asymmetrically supervised by the ground-truth target modality images x_i^T and class labels y_i. As is represented in Eq. (4.1), the loss term $\mathcal{L}_{pixel}$ is controlled by a binary factor corresponding to class labels y_i, which enforces the training of the translator focus on genuine face images only and thereby achieves the goal of asymmetric translation. N, N_g, and i denote the total number of samples, the number of genuine face samples, and the sample index. For simplification, Eq. (4.1) can be rewritten into Eq. (4.2), where G is the set of indexes of genuine face samples.

$$\begin{aligned} \mathcal{L}_{pixel} &= \frac{1}{N_g}\sum_{i=1}^{N}(1-y_i) \parallel {x'}_i^T - x_i^T \parallel, \\ y_i &= \begin{cases} 0, & genuine, \\ 1, & attack. \end{cases} \end{aligned} \tag{4.1}$$

$$\mathcal{L}_{pixel} = \frac{1}{N_g}\sum_{i\in G} \parallel {x'}_i^T - x_i^T \parallel . \tag{4.2}$$

In the latent space, the discrepancy between genuine face and attack samples is explicitly enlarged with a loss term adapted from the supervised contrastive loss [116]. We use G and A to denote the sets of indexes of genuine and attack samples.

The sizes of the two sets can be represented by $|G|$ and $|A|$. Each genuine face sample can be paired with other genuine ones to generate $|G|-1$ positive pairs and with attacks to generate $|A|$ negative pairs. In our expected latent space, we wish the similarity of positive pairs is greater than negative pairs to ensure the consistency of genuine samples and the discrepancy between genuine and attack samples. The disparity with expectation can be measured by an asymmetric supervised contrastive loss. For each genuine face sample z_i, the loss value $\mathcal{L}_{latent,i}$ is defined as

$$\mathcal{L}_{latent,i} = \frac{1}{|G|-1} \sum_{g \in G, g \neq i} -\log \frac{\exp(z_i \cdot z_g/\tau)}{\sum\limits_{a \in A} \exp(z_i \cdot z_a/\tau)}. \tag{4.3}$$

z_i, z_g are L_2-normalized latent features of genuine face samples with index i and g. z_a is L_2-normalized latent feature of attack samples with index a. The temperature τ is a constant hyperparameter. The inner product of feature vectors in latent space is used to measure the similarity between two data samples. The relativeness is formulated by contrasting each positive pair with the sum of all negative pairs. Since the range of logarithmic term is $(-\infty, \infty)$, we truncate it at a fixed value c to make it stable and compatible with L_{pixel}. The loss of overall genuine samples is represented as

$$\mathcal{L}_{latent} = \frac{1}{|G|-1} \sum_{i \in G} \sum_{g \in G, g \neq i} \max(-\log \frac{\exp(z_i \cdot z_g/\tau)}{\sum\limits_{a \in A} \exp(z_i \cdot z_a/\tau)}, c). \tag{4.4}$$

By combining the loss term in both pixel and latent space, we get the complete loss function for asymmetric modality translation as in Eq. (4.4). Supervised by which the CNN-based translator is trained to meet our expected functionalities.

$$\mathcal{L}_{AMT} = \lambda_1 \cdot \mathcal{L}_{pixel} + \lambda_2 \cdot \mathcal{L}_{latent}. \tag{4.5}$$

4.2.3 Modality Fusion and Discrimination

As shown in Figure 4.2, the translated image $x_i^{'T}$ is fused with the ground-truth target modality image x_i^T captured by camera sensor. We have implemented two alternative operations (concatenation and subtraction) for the fusion layer. The subtraction operation aggregates information by directly computing the absolute

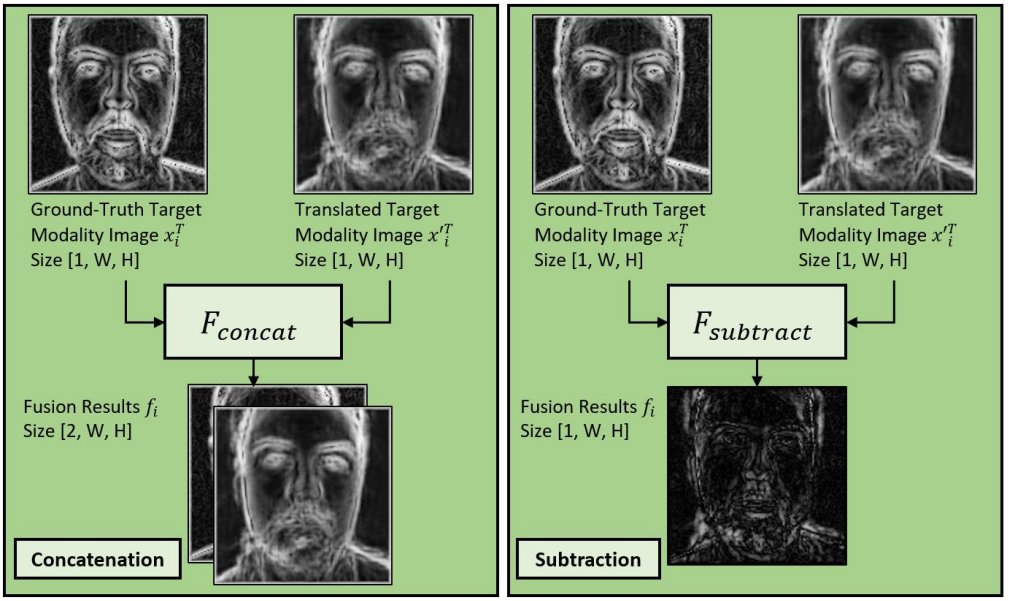


FIGURE 4.3: The figure shows the illustration of two fusion operations. F_{concat} and $F_{subtract}$ denote the concatenation and subtraction operation respectively. W and H denote the width and height of the image.

value of the pixel-level difference. The operation result is a straightforward measurement of the similarity between two images. We believe it is effective information for the discriminator to distinguish spoofing face images from genuine ones. The concatenation operation aggregates two images along the channel axis, which is commonly used in the general I2I task [119]. The information of two images is aggregated according to the spatial position and fully kept in the operation result, which provides the discriminator detail information to learn high-level discrepancies between genuine and spoofing face images.

The mathematical representations are shown as Eq. (4.6) and Eq. (4.7). $[\cdot;\cdot]$ denotes the concatenation of two images along the channel axis. $||\cdot||$ and $-$ are pixel-wise operations that denote the calculation of absolute value and subtraction, respectively. For subtraction one, we replicate the fusion result along the channel axis to make the shape the same as the concatenation one.

$$f_i = F_{concat}(x'^T_i, x^T_i) = [x'^T_i; x^T_i]. \tag{4.6}$$

$$f_i = F_{subtract}(x'^T_i, x^T_i) = ||x'^T_i - x^T_i||. \tag{4.7}$$

The details about the two fusion operations are shown in Figure 4.3. Based on the experimental results in Chapter 4.4.5, we finally select the concatenation one as the fusion layer. The fusion result f_i is fed as the input to a discriminator, which outputs a patch map d_i that can be interpreted as the patch-wise discrepancy. The output d_i of the discriminator is supervised with binary cross-entropy (BCE) loss in a pixel-wise manner similar to [66]. The discrimination loss is shown as

$$\begin{aligned} \mathcal{L}_{dis} &= \frac{1}{N}\sum_{i=1}^{N} -(y_i \log(d_i) + (1-y_i)\log(1-d_i)), \\ y_i &= \begin{cases} 0, & genuine, \\ 1, & attack. \end{cases} \end{aligned} \tag{4.8}$$

Following [66], the network architecture of the discriminator is implemented with the block of DenseNet [101]. Since the DC module is jointly trained with the AMT module in an end-to-end way, the total loss function is represented as

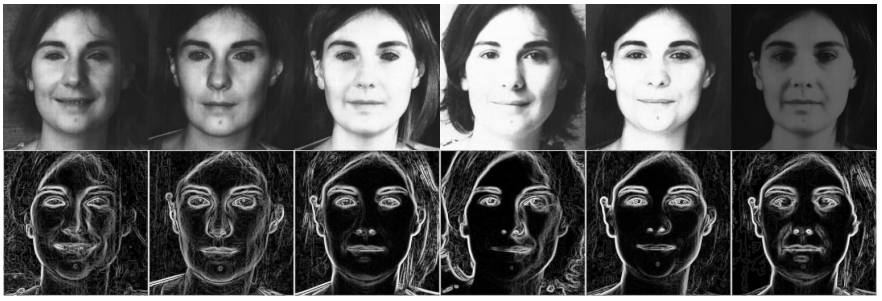
$$\begin{aligned} \mathcal{L}_{total} &= \mathcal{L}_{AMT} + \lambda_3 \cdot \mathcal{L}_{dis} \\ &= \lambda_1 \cdot \mathcal{L}_{pixel} + \lambda_2 \cdot \mathcal{L}_{latent} + \lambda_3 \cdot \mathcal{L}_{dis}. \end{aligned} \tag{4.9}$$

4.2.4 PLGF-based Illumination Normalization

As is known, VIS-based face PAD methods are sensitive to the variation of illumination conditions [3]. This problem also exists in bi-modality scenarios, where some modalities like VIS or NIR are employed. The illumination variation is even tougher trouble for the training of our asymmetric modality translator. For example, under different illuminations, the same image of depth modality may correspond to different VIS images of considerable intensity variation.

To reduce the impact of illumination variation, we devise an IN module based on PLGF descriptor [122], which is initially proposed for illumination-invariant face recognition. As shown in Figure 4.2, images of different modalities are selectively processed by predefined convolution masks to filter the illumination components. Only images of illumination-sensitive modalities such as VIS and NIR are processed in our design, while images of other modalities skip this normalization. The PLGF

(A) Genuine Face Images



(B) Printed Face Images

FIGURE 4.4: The figure shows the visualization results of illumination normalization. The first row of (A) shows the raw VIS images of genuine faces in MSSPOOF dataset that captured under different illumination conditions. The second row of (A) shows corresponding VIS images processed by IN module (IN-VIS). Sub-figure (B) shows VIS and IN-VIS images of attack samples.

descriptor is represented as

$$x_i = \arctan(\sqrt{(\frac{r_i * M_x}{r_i})^2 + (\frac{r_i * M_y}{r_i})^2}). \tag{4.10}$$

r_n and x_n are the raw and processed images respectively, M_x and M_y are two filter masks, * is convolution operation and all the other mathematical operations in Eq. (4.10) are pixel-wise. The representations of two filter masks are shown as

$$M_x(p,q) = \begin{cases} \dfrac{\cos(\arctan 2(q,p))}{p^2+q^2}, & (p^2+q^2) > 0, \\ 0, & otherwise, \end{cases} \tag{4.11}$$

$$M_y(p,q) = \begin{cases} \dfrac{\sin(\arctan 2(q/p))}{p^2+q^2}, & (p^2+q^2) > 0, \\ 0, & otherwise. \end{cases} \tag{4.12}$$

p and q are indexes denoting the relative position to the center. In our IN module, the size of masks are set to 5, therefore the range of p, q is $-2 \leq p, q \leq 2$. The visualization results are shown in Figure 4.4.

TABLE 4.1: List of Dataset Information

Dataset	Modality Settings	Types of Attacks
WMCA [8]	VIS(V)-NIR(I) Thermal(T)-VIS(V) Thermal(T)-NIR(I)	Glasses, Fake Head, Print, Replay, Rigid Mask, Flexible Mask, Paper Mask
CASIA-SURF [6, 99]	VIS(V)-NIR(I) Depth(D)-VIS(V) Depth(D)-NIR(I)	Print(bent, eyes cropped) Print(bent, eyes-nose cropped) Print(bent, eyes-nose-mouth cropped) Print(still, eyes cropped) Print(still, eyes-nose cropped) Print(still, eyes-nose-mouth cropped)
MSSPOOF [5]	VIS(V)-NIR(I)	Print

4.3 Experimental Setup

4.3.1 Dataset Information

To verify the effectiveness of our proposed method, we conducted extensive experiments on 3 publicly available datasets: Wide-Multi Channel Presentation Attack (WMCA) [8], CASIA-SURF [6, 99], and Multispectral-Spoof Database (MSSPOOF) [5], which are commonly used for multi-modality face anti-spoofing research. The cause of choosing these datasets is that they vary in multiple aspects, such as the number of samples, data modality, attack type, specification of the sensor device, and even preprocessing schemes. We believe this will provide a more general evaluation.

WMCA dataset contains images of VIS (V for short), NIR (I for short), thermal (T for short), and depth (D for short) modalities. Grouping each two of them, there are 6 combinations. In this work, we evaluated our method mainly under WMCA (V-I), WMCA (T-V), and WMCA (T-I) settings. Each data sample in CAISA-SURF dataset has 3 images of V, I, and D modality. We pair each two of them and establish 3 bi-modality settings referred to as CASIA-SURF (V-I), CASIA-SURF (D-V), and CASIA-SURF (D-I). On MSSPOOF dataset, we established the modality setting denoted as MSSPOOF (V-I). The modality and type of attack information of the three datasets are listed in Table 4.1.

4.3.2 Evaluation Protocols

1) Grand-Test Protocol:
Grand-test is a basic protocol to verify the effectiveness of face PAD methods. It simulates an ideal situation to evaluate the overall performance. For this protocol, all the *train*, *dev* and *test* sets contain genuine samples and all types of attack samples.

2) Unseen-Attack Protocol:
Unseen-attack evaluation is more realistic in practical scenarios. Unlike grand-test protocol, specific attack types evaluated at the test stage are not available at the train and development stage. Leave-one-out (LOO) protocol is commonly used for unseen-attack evaluation. For each sub-protocol, one specific type of attack is left out from the *train* and *dev* set, while the *test* set only remains genuine samples and the left type of attack ones. For the CASIA-SURF dataset, in addition to the LOO protocol, we also evaluated our methods under the protocol used in [6, 99], which we noted Leave-three-out (LTO). We have not conducted unseen-attack evaluations on the MSSPOOF dataset due to the small population of data samples and limited attack types.

3) Cross-Illumination Protocol:
Cross-illumination evaluation is a necessary step to test the robustness of face PAD systems under varying illumination conditions. Similar to unseen-attack evaluation, we also adopt the LOO protocol for cross-illumination evaluation. Specifically, for each sub-protocol, we leave samples under one illumination out from the *train* and *dev* subsets, while in the *test* set, only the samples under the left illumination condition are evaluated. We conducted cross-illumination evaluations on WMCA dataset because it covers samples under various illumination conditions and provides illumination labels.

4) Cross-Dataset Protocol:
Cross-dataset evaluation is the most challenging scenario, which is used to evaluate the robustness of the trained model under domain shift caused by multiple factors. V-I modality settings of the three datasets are used for cross-dataset experiments. For each dataset, we train a model on its own *train* set and evaluate the *the same* model on the *test* sets of other two datasets, in addition to its own *test* set. Therefore, there are 6 sub-protocols which are referred to as $M \rightarrow W$, $W \rightarrow M$,

$C \rightarrow W$, $W \rightarrow C$, $C \rightarrow M$, and $M \rightarrow C$. W, M, and C denote WMCA, MSSPOOF, and CASIA-SURF, respectively.

4.3.3 Evaluation Metrics

Various metrics have been proposed for performance evaluation in face PAD research, and different metrics are usually used in different works for performance reports. For comprehensive comparisons, we used Attack Presentation Classification Error Rate (APCER), Bonafide Presentation Classification Error Rate (BPCER), Average Classification Error Rate (ACER), Equal Error Rate (EER), TDR@FDR=1%, and Area Under Curve (AUC) in our experiments. Larger values are better for AUC and TDR@FDR=1%, while smaller values are better for others.

Face PAD is a typical binary classification task, for which the threshold is usually used for decision making. AUC measures the overall performance over different thresholds. APCER, BPCER, ACER, EER, and TDR@FDR=1% are metrics to evaluate performance at a fixed threshold. APCER measures the error rate that the system classifies attack samples as genuine ones. In complementary, BPCER measures the error rate that the system classifies genuine samples as attack ones. ACER is the average of APCER and BPCER and is used to evaluate the overall performance. Following [8], we report APCER, BPCER, and ACER at the threshold where BPCER=1% on *dev* set. While AUC and TDR@FDR=1% are calculated directly on *test* set as in [6, 99].

4.3.4 Baseline Methods

To verify the effectiveness of our proposed method and compare it with SOTA methods under fair experimental settings, we have also re-implemented 2 multi-modality-based face PAD methods [9, 102] as our baselines. We also compared with other multi-modality-based methods [6, 8, 99] when under comparable settings.

1) MC-PixBiS:
MC-PixBiS[9] is a multi-modality face PAD method extended from DeepPixBiS [66], which achieves good performance under different multi-modality face PAD benchmarks. This method concatenates images of different modalities as the input

to a discriminator based on DenseNet [101]. The training process is end-to-end and supervised by pixel-wise binary and binary labels. In our experiments, we set the parameters of this method as the same to [9] and denote it as MC-PixBiS.

2) MM-CDCN:
MM-CDCN [102] is based on CDC proposed in [48], which is originally designed to address VIS-based face PAD problem. Compared to conventional convolution, CDC learns more detailed features, which is suitable for the face PAD task. In our experiment, we adopt the input-level fusion like MC-PixBiS and denote it as MM-CDCN.

3) Other Methods:
Besides aforementioned methods implemented by us, we also compared performance with other methods such as RDWT+Haralick [23], IQM [18, 19]+LBP[22], MC-CNN [8], Single-Scale SEF [6] and Multi-Scale SEF[99] when evaluated under comparable experimental settings.

4.3.5 Implementation Details

Since the quantity of attack samples is much larger than genuine ones in all three datasets, we balanced the *train* set by uniformly upsampling the genuine ones with factors 5, 4, and 3 on CASIA-SURF, WMCA, and MSSPOOF to make the amounts of genuine and attack samples in *train* set comparable. Cropped face images were resized to 128×128 before being fed into the proposed bi-modality framework. The λ_1, λ_2 and λ_3 in Eq.(4.9) are set as 5×10^{-1}, 1×10^{-3} and 1, respectively. The training process is optimized by Adam optimizer with the mini-batch size of 32, and the learning rate, which is initially set as 10^{-4}, then decreases by half for every 10 epochs. According to the number of data samples in the training set, the model is trained for up to 20, 30, and 120 epochs on CASIA-SURF, WMCA, and MSSPOOF datasets. The proposed framework and benchmark methods are implemented based on PyTorch version 1.7.0, and all our experiments are conducted on GPUs with CUDA version 10.1.

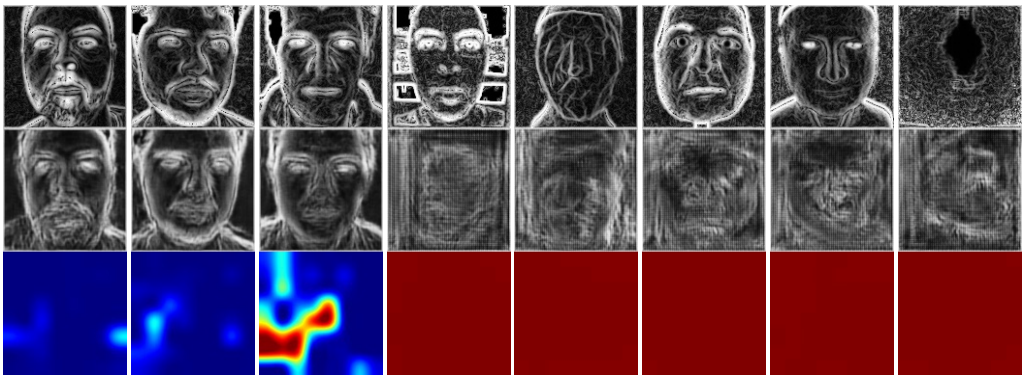


FIGURE 4.5: The figure shows the visualization of images under WMCA (T-I) setting. Columns (A)-(C) are images of genuine face samples, and columns (D)-(H) are images of attack samples. From top to bottom row are images of ground-truth IN-NIR, IN-NIR that translated from thermal images and corresponding patch maps. The patch maps are colorized as heat maps, where the red region indicates anomaly.

4.4 Results and Analysis

4.4.1 Grand-Test Experiments

In order to verify the effectiveness of our proposed method, we conducted extensive experiments with 7 modality settings from 3 datasets and compared the performance with SOTA methods in terms of AUC, TDR@FDR=1% APCER, BPCER, and ACER.

1) Results on WMCA:
We firstly compared our method with two baseline methods under the same modality settings. As shown in Table 4.2, MC-PixBiS has comparable performance to MM-CDCN under the T-I setting and better performance under both V-I and T-V settings. Our method clearly outperforms two baseline methods under all settings. In specific, our method reduces the ACER by 3.69%, 1.04%, and 2.05% under V-I, T-V, and T-I settings severally, compared to MC-PixBiS. By comparing performance over different modality settings, we find that all three methods achieve their best performance under T-I settings, which indicates the importance of modality selection. Besides, we also compared other methods which use additional modalities. From Table 4.3, we can see that our method significantly outperforms RDWT+Haralick and IQM+LBP methods under different bi-modality

TABLE 4.2: Grand-Test Experimental Results on WMCA (Same Modality)

Modality Setting	Method	Metrics(%)				
		AUC	TDR@FDR=1%	APCER	BPCER	ACER
		↑	↑	↓	↓	↓
V-I	MM-CDCN [102]	97.41	81.70	17.38	1.15	9.26
	MC-PixBiS [9]	99.92	97.64	9.76	**0.00**	4.88
	Ours [112]	**99.94**	**98.64**	**1.41**	0.97	**1.19**
T-V	MM-CDCN [102]	98.83	92.34	7.14	1.27	4.20
	MC-PixBiS [9]	98.64	96.38	4.10	**0.89**	2.50
	Ours [112]	**99.92**	**98.99**	**0.86**	2.05	**1.46**
T-I	MM-CDCN [102]	99.20	96.05	3.94	1.08	2.51
	MC-PixBiS [9]	99.12	96.34	3.84	0.71	2.28
	Ours [112]	**99.99**	**99.71**	**0.45**	**0.02**	**0.23**

TABLE 4.3: Grand-Test Experimental Results on WMCA (Different Modality)

Method	Metrics(%)				
	AUC	TDR@FDR=1%	APCER	BPCER	ACER
	↑	↑	↓	↓	↓
RDWT+Haralick (V-I-T-D) [8]	/	/	6.39	0.49	3.44
IQM+LBP (V-I-T-D) [8]	/	/	13.92	1.17	7.54
MC-CNN (V-I-T-D) [8]	/	/	0.60	**0.00**	0.30
Ours (V-I) [112]	99.94	98.64	1.41	0.97	1.19
Ours (T-V) [112]	99.92	98.99	0.86	2.05	1.46
Ours (T-I) [112]	**99.99**	**99.71**	**0.45**	0.02	**0.23**

settings. Moreover, under the T-I setting, our method performs competitively to MC-CNN while using fewer modalities. Some visualizations of our method are shown in Figure 4.5.

2) Results on CASIA-SURF:

From experiments on CASIA-SURF, we can see that our method consistently performs better than baseline methods under all three settings, as in Table 4.4. Especially for the V-I setting, where both baseline methods perform poorly, our method outperforms them by distinct margins in terms of TDR@FDR=1%, ACER, and APCER. Performance on CASIA-SURF shows the effectiveness of our method under challenging settings.

TABLE 4.4: Grand-Test Experimental Results on CASIA-SURF

Modality Setting	Method	Metrics(%)				
		AUC ↑	TDR@FDR=1% ↑	APCER ↓	BPCER ↓	ACER ↓
V-I	MM-CDCN [102]	99.24	89.45	11.83	0.80	6.32
	MC-PixBiS [9]	93.52	73.06	26.23	0.98	13.60
	Ours [112]	**99.79**	**96.04**	**3.94**	**0.78**	**2.36**
D-V	MM-CDCN [102]	99.88	98.28	1.22	1.48	1.35
	MC-PixBiS [9]	99.36	94.58	3.43	1.84	2.63
	Ours [112]	**99.97**	**99.45**	**0.45**	**1.02**	**0.74**
D-I	MM-CDCN [102]	99.73	95.74	2.38	1.78	2.08
	MC-PixBiS [9]	99.30	87.84	7.63	**1.61**	4.62
	Ours [112]	**99.97**	**99.66**	**0.20**	1.73	**0.96**

TABLE 4.5: Grand-Test Experimental Results on MSSPOOF

Modality Setting	Method	Metrics(%)				
		AUC ↑	TDR@FDR=1% ↑	APCER ↓	BPCER ↓	ACER ↓
V-I	MM-CDCN [102]	99.81	97.22	1.86	1.92	1.89
	MC-PixBiS [9]	99.97	98.61	**0.00**	2.40	1.20
	Ours [112]	**100.00**	**100.00**	**0.00**	**0.96**	**0.48**

3) Results on MSSPOOF:

To verify the effectiveness of our method on small datasets, we conducted experiments on MSSPOOF as well. According to the results in Table 4.5, we see that our method still outperforms baseline methods. Moreover, it achieves perfect performance in terms of AUC, TDR@FDR=1%, and APCER, demonstrating that our method is valid even with limited training data.

4.4.2 Cross-Illumination Experiments

In order to evaluate the robustness of our method under varied illumination conditions, we conducted cross-illumination experiments on the WMCA dataset V-I modality setting, where both modalities are sensitive to illumination variations. Data samples in the WMCA [8] were captured under 7 different illumination conditions, while there are no genuine face samples under illumination 4. Therefore, there are only 6 sub-protocols with the LOO. From Table 4.6, we can see that our method stably performs better than two baseline methods by more than 7% in terms of ACER. Although MC-PixBiS achieves comparable performance in terms of AUC, the ACER is severely degraded. It is because the threshold of ACER is determined by *dev* set and different illumination conditions of *dev* and *test* cause the shift of the decision threshold.

TABLE 4.6: Cross-Illumination (LOO) Experimental Results on WMCA

Modality Setting	Metrics(%)	Method	Type of Illumination						Overall
			1	2	3	5	6	7	
V-I	ACER (↓)	MM-CDCN [102]	8.46	17.35	13.71	9.79	9.53	5.43	10.71±4.20
		MC-PixBiS [9]	2.88	18.27	14.78	5.36	2.71	4.47	8.08±6.71
		Ours [112]	0.74	1.75	0.14	0.47	1.54	0.08	**0.79±0.71**
	AUC (↑)	MM-CDCN [102]	98.36	90.66	96.50	97.71	96.20	99.29	96.45±3.06
		MC-PixBiS [9]	99.96	99.38	98.60	100.00	99.40	99.51	99.48±0.51
		Ours [112]	99.98	99.85	100.00	100.00	99.91	100.00	**99.96±0.06**
	TDR@FDR=1% (↑)	MM-CDCN [102]	80.59	69.97	79.80	87.45	81.52	90.85	81.70±7.20
		MC-PixBiS [9]	98.59	93.04	92.38	100.00	97.21	92.68	95.65±3.36
		Ours [112]	99.79	97.22	100.00	100.00	98.10	100.00	**99.19±1.22**

TABLE 4.7: Unseen-Attack (LOO) Experimental Results on WMCA (Same Modality)

Modality Setting	Metrics(%)	Method	Type of Attack							Overall
			Glasses	Fake Head	Print	Replay	Rigid Mask	Flexible Mask	Paper Mask	
			1	2	3	4	5	6	7	
V-I	ACER (↓)	MM-CDCN [102]	41.55	44.77	0.98	0.92	19.34	19.23	26.77	21.94±17.44
		MC-PixBiS [9]	45.45	48.14	0.02	0.00	0.87	19.16	4.44	16.87±21.52
		Ours [112]	35.66	0.83	0.03	0.07	0.20	7.13	0.44	**6.34±13.18**
	AUC (↑)	MM-CDCN [102]	85.24	97.01	99.99	99.93	96.13	93.98	91.56	94.83±5.20
		MC-PixBiS [9]	73.90	91.34	100.00	100.00	99.99	96.03	99.85	94.44±9.63
		Ours [112]	81.68	99.94	100.00	100.00	100.00	98.52	99.96	**97.16±6.85**
	TDR@FDR=1% (↑)	MM-CDCN [102]	10.27	37.65	100.00	99.00	44.16	36.35	28.03	50.78±34.95
		MC-PixBiS [9]	28.36	10.82	100.00	100.00	99.51	49.26	97.38	69.33±38.91
		Ours [112]	16.45	98.71	100.00	100.00	100.00	80.91	100.00	**85.15±31.10**
T-V	ACER (↓)	MM-CDCN [102]	44.73	0.67	0.37	0.30	6.20	2.65	11.59	9.50±16.07
		MC-PixBiS [9]	45.50	0.18	0.00	0.43	2.85	0.46	6.79	8.03±16.70
		Ours [112]	36.84	0.51	0.43	0.22	1.33	0.36	1.57	**5.89±13.66**
	AUC (↑)	MM-CDCN [102]	68.42	99.91	100.00	100.00	98.82	99.63	95.61	94.63±11.66
		MC-PixBiS [9]	49.25	100.00	100.00	99.99	99.58	99.93	98.23	92.43±19.05
		Ours [112]	64.89	99.88	100.00	100.00	99.96	100.00	99.83	**94.94±13.25**
	TDR@FDR=1% (↑)	MM-CDCN [102]	9.55	99.76	100.00	100.00	88.73	95.58	74.00	81.09±32.91
		MC-PixBiS [9]	2.73	99.88	100.00	100.00	94.04	99.97	86.62	83.32±35.89
		Ours [112]	10.82	100.00	100.00	100.00	98.78	99.82	97.46	**86.70±33.47**
T-I	ACER (↓)	MM-CDCN [102]	38.89	1.14	0.47	0.44	5.45	1.77	5.28	7.63±13.95
		MC-PixBiS [9]	39.14	0.62	0.26	0.53	1.22	1.10	3.42	6.61±14.38
		Ours [112]	37.74	0.43	0.07	0.02	0.66	0.31	0.00	**5.60±14.17**
	AUC (↑)	MM-CDCN [102]	74.29	99.96	100.00	100.00	98.92	99.88	98.51	95.94±9.56
		MC-PixBiS [9]	49.39	99.97	100.00	100.00	99.81	99.97	99.50	92.66±19.08
		Ours [112]	78.30	99.97	100.00	100.00	100.00	100.00	100.00	**96.90±8.20**
	TDR@FDR=1% (↑)	MM-CDCN [102]	17.45	99.29	100.00	100.00	90.82	97.84	89.89	85.04±30.11
		MC-PixBiS [9]	18.36	99.88	100.00	100.00	90.86	98.74	89.58	85.35±29.88
		Ours [112]	15.73	100.00	100.00	100.00	100.00	99.91	100.00	**87.95±31.85**

4.4.3 Unseen-Attack Experiments

In practical scenarios, the deployed face PAD system may always encounter novel attacks unseen by the system designers during the model training phase. Therefore, we did experiments under unseen-attack protocols to validate the robustness of our method to zero-shot attacks.

1) Results on WMCA:

For a fair comparison, we firstly compared our method with re-implemented baselines under three modality settings of the WMCA dataset with the LOO protocol. As is shown in Table 4.7, our method has achieved better results under all evaluation settings. Specifically, our method outperforms baseline methods by more than 10.53%, 2.12%, and 1.01% for mean ACER; by 2.33%, 0.31%, and 0.96% for mean AUC; by 15.82%, 3.38%, and 2.91% for mean TDR@FDR=1%, under

TABLE 4.8: Unseen-Attack (LOO) Experimental Results on WMCA (Different Modality)

Metrics	Method	Type of Attack							Overall
		Glasses	Fake Head	Print	Replay	Rigid Mask	Flexible Mask	Paper Mask	
		1	2	3	4	5	6	7	
ACER (↓)	RDWT+Haralick (V-I-T-D) [8]	48.85	3.16	0.00	5.77	7.65	14.05	2.25	11.68±17.01
	IQM+LBP (V-I-T-D) [8]	50.86	2.38	2.30	0.84	14.27	28.58	16.34	16.51±18.16
	MC-CNN (V-I-T-D) [8]	42.14	0.00	0.00	0.12	0.75	2.52	0.35	6.55±15.72
	Ours (V-I) [112]	35.66	0.83	0.03	0.07	0.20	7.13	0.44	6.34±13.18
	Ours (T-V) [112]	36.84	0.51	0.43	0.22	1.33	0.36	1.57	5.89±13.66
	Ours (T-I) [112]	37.74	0.43	0.07	0.02	0.66	0.31	0.00	**5.60±14.17**

TABLE 4.9: Unseen-Attack (LOO) Experimental Results On CASIA-SURF

Modality Setting	Metrics(%)	Method	Type of Attack						Overall
			Eyes-Still	Eyes-Bent	Eyes-Nose-Still	Eyes-Nose-Bent	Eyes-Nose-Mouth-Still	Eyes-Nose-Mouth-Bent	
			1	2	3	4	5	6	
D-I	ACER (↓)	MM-CDCN [102]	2.02	2.53	1.71	2.18	1.56	6.41	2.74±1.83
		MC-PixBiS [9]	4.43	6.67	3.23	4.51	3.14	7.13	4.85±1.69
		Ours [112]	1.34	1.26	0.59	0.86	0.67	1.17	**0.98±0.32**
	AUC (↑)	MM-CDCN [102]	99.78	99.67	99.81	99.70	99.81	98.90	99.61±0.35
		MC-PixBiS [9]	99.15	97.89	99.22	98.90	99.61	96.94	98.62±1.01
		Ours [112]	99.92	99.88	99.97	99.95	99.97	99.90	**99.93±0.04**
	TDR@FDR=1% (↑)	MM-CDCN [102]	95.83	94.74	96.99	95.28	97.32	78.05	93.04±7.41
		MC-PixBiS[9]	79.87	79.27	90.87	83.83	91.20	72.18	82.87±7.36
		Ours [112]	97.86	98.19	99.75	99.35	99.52	98.83	**98.92±0.76**

V-I, T-V, and T-I respectively. Besides, we see that our method performs stably over different modality settings. In contrast, the performances of baseline methods degrade severely under the V-I setting, where both modalities (VIS and NIR image) are sensitive to the environment illumination. Benefiting from IN module, our method is significantly more effective to fake head, flexible mask, and paper mask attack at unseen attack scenario, compared with baselines. To further verify the superiority of our method, in Table 4.8, we also compared with SOTA methods under different modality settings, even though they used additional modality information. United with different modalities, our method performs better than RDWT+Haralick, IQM+LBP, and MC-CNN. Especially, our method under the T-I setting outperforms SOTA on glasses attacks and has achieved nearly perfect performance on other attacks in terms of ACER.

2) Results on CASIA-SURF:

On the CASIA-SURF dataset, we evaluated our method under both the LOO and LTO protocols. For the LOO protocol, as shown in Table 4.9, our method outperforms two baseline methods under all sub-protocols, especially for bent photo attacks with eye-nose-mouth cropping. The mean ACER, mean AUC, and mean TDR@FDR=1% of our method are 0.98%, 99.93%, and 98.92%, which are quite close to the performance under the grand-test setting. For LTO protocol, in addition to our implemented baseline methods, we also compared the proposed method with two benchmark methods on CASIA-SURF, referred to as Single-Scale SEF [6] and Multi-Scale SEF [99], under best performing modality D-I. As shown in

TABLE 4.10: Unseen-Attack (LTO) Experimental Results on CASIA-SURF

Modality Setting	Method	Metrics(%)				
		AUC	TDR@FDR=1%	APCER	BPCER	ACER
		↑	↑	↓	↓	↓
D-I	Single-Scale SEF [6]	/	/	1.5	8.4	4.9
	Multi-Scale SEF [99]	/	/	2.0	**0.3**	1.1
	MM-CDCN [102]	99.74	95.46	2.31	2.15	2.23
	MC-PixBiS [9]	98.95	80.56	6.34	2.79	4.56
	Ours [112]	**99.95**	**99.46**	**0.54**	0.99	**0.77**

TABLE 4.11: Cross-Dataset Experimental Results

Method	$M \rightarrow W$				$W \rightarrow M$			
	Intra-Performance		Inter-Performance		Intra-Performance		Inter-Performance	
	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑
MM-CDCN [102]	1.89	99.81	58.41	39.34	7.51	97.41	48.05	54.75
MC-PixBiS [9]	1.42	99.97	44.46	55.20	1.54	99.92	47.22	58.43
Ours [112]	**0.47**	**100.00**	**35.88**	**69.77**	**1.15**	**99.94**	**24.08**	**86.76**
Method	$C \rightarrow W$				$W \rightarrow C$			
	Intra-Performance		Inter-Performance		Intra-Performance		Inter-Performance	
	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑
MM-CDCN [102]	4.01	99.24	54.06	44.83	7.51	97.41	**38.54**	**65.40**
MC-PixBiS [9]	11.63	93.52	**49.88**	**52.51**	15.4	99.92	46.30	52.14
Ours [112]	**2.00**	**99.79**	52.59	48.41	**1.15**	**99.94**	53.77	52.84
Method	$C \rightarrow M$				$M \rightarrow C$			
	Intra-Performance		Inter-Performance		Intra-Performance		Inter-Performance	
	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑
MM-CDCN [102]	4.01	99.24	**36.01**	**72.64**	1.89	99.81	49.40	51.81
MC-PixBiS [9]	11.63	93.52	69.19	32.05	1.42	99.97	**41.60**	**61.91**
Ours [112]	**2.00**	**99.79**	42.27	59.49	**0.47**	**100.00**	44.30	57.44

Table 4.10, our method excels Single-Scale SEF and MC-PixBiS by a clear margin of about 4% in ACER. Compared with Multi-Scale SEF, our method achieves comparable results.

4.4.4 Cross-Dataset Experiments

We also conducted experiments under cross-dataset protocols to test the feasibility of transferring the trained model to a novel domain without fine-tuning it with newly available data. For all methods, we employed the best-performing model under grand-test experiments. We chose EER and AUC as metrics for performance evaluation, and both intra-dataset and inter-dataset performance were reported based on the same model. As shown in Table 4.11, all three evaluated methods degrade severely, and no method outstands for all sub-protocols. We observe our method consistently performs better on cross-evaluation between WMCA and MSSPOOF compared with baseline methods. However, it does not perform well

for $C - W$ and $C - M$. We believe it is because the data preprocessing conducted on CASIA-SURF is quite different from WMCA and MSSPOOF. For the $M - W$ setting, although we applied the same preprocessing procedures, the performance is still far from expected. The cause is that data samples in two datasets are captured with camera sensors of different specifications and under different environments. From our experiments, we find that the cross-dataset generalization problem on multi-modality face PAD research may not be easier than single VIS-modality ones. The specifications of sensors and data preprocessing schemes should be unified to protect the performance from being severely degraded when multi-modality-based face PAD systems land for practical application.

4.4.5 Ablation Study

To better analyze our proposed method, we did a series of ablation experiments to study the contribution of different components and their scalability. All the experiments are conducted under the CASIA-SURF (D-I) and WMCA (T-I) settings, where our methods show superiority.

1) Contribution of Different Components:
We firstly conducted module ablation tests to study the necessity of different components. As shown in Table 4.12, both the IN module and AMT loss contribute to improving the performance under both evaluation settings. Without the IN module, the performance of our method degrades 1.54%, 1.12% in terms of TDR@FDR=1%, ACER under the WMCA (T-I) setting. While under the CASIA-SURF (D-I) setting, the degradation is about 1.62% and 0.45%. Without the AMT loss, there is a degradation of 3.65%, 2.17% and 6.75%, 2.54% under the WMCA (T-I) and CASIA-SURF (D-I), respectively. Figure 4.6 shows the visualization of ground-truth and translated images under WMCA (T-I) settings. Comparing the third and fifth row with the second and fourth row, we can see that the genuine face images are better translated from thermal modality to NIR modality, while attack samples fail. It shows the trained asymmetric modality translator meets our expectations. Compared to translated raw NIR images, translated NIR images which are processed by IN module (IN-NIR) between different attacks are of smaller difference.

TABLE 4.12: Experimental Results with Different Components

Modality Setting	Variants	Metrics(%)				
		AUC ↑	TDR@FDR=1% ↑	APCER ↓	BPCER ↓	ACER ↓
WMCA(T-I)	w/o IN [112]	99.91	98.17	1.57	1.13	1.35
	w/o $\mathcal{L}_{AMT}$ [112]	99.61	96.06	3.02	1.77	2.40
	full [112]	**99.99**	**99.71**	**0.45**	**0.02**	**0.23**
CASIA-SURF(D-I)	w/o IN [112]	99.84	98.04	1.60	**1.21**	1.41
	w/o $\mathcal{L}_{AMT}$ [112]	99.56	92.91	5.58	1.41	3.50
	full [112]	**99.97**	**99.66**	**0.20**	1.73	**0.96**

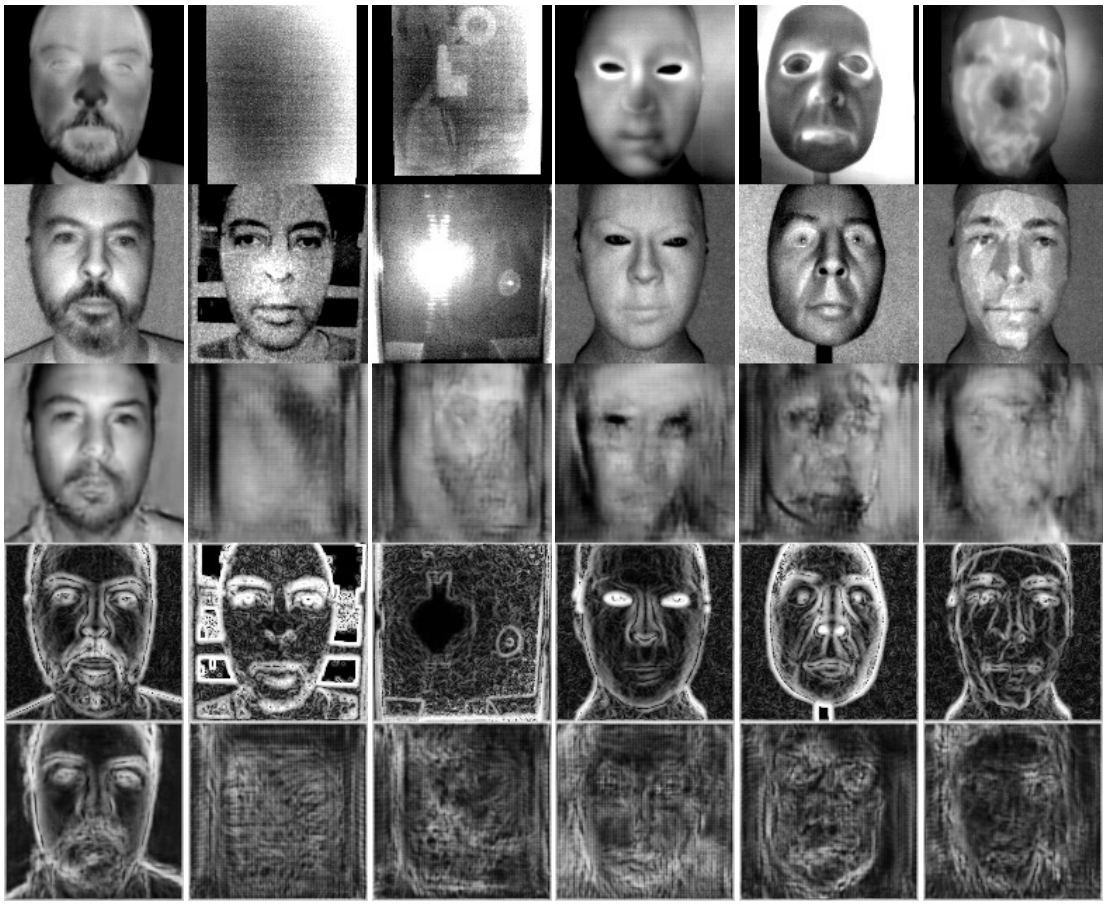
TABLE 4.13: Experimental Results with Different Fusion Operations

Modality Setting	Variants	Metrics(%)				
		AUC ↑	TDR@FDR=1% ↑	APCER ↓	BPCER ↓	ACER ↓
WMCA(T-I)	subtraction [112]	99.46	97.07	2.85	1.25	2.05
	concatenation [112]	**99.99**	**99.71**	**0.45**	**0.02**	**0.23**
CASIA-SURF(D-I)	subtraction [112]	99.92	98.56	1.11	**1.24**	1.18
	concatenation [112]	**99.97**	**99.66**	**0.20**	1.73	**0.96**

2) Comparison Between Different Fusion Operations:
We compared two operations for fusion, computing the difference of translated images with the ground truth and concatenating them. As shown in Table 4.13, under CASIA-SURF (D-I), both fusion operations perform well, and there is no obvious difference, while under the WMCA (T-I) setting, the concatenation one stands out. Therefore, we experimentally chose the concatenation one for our framework.

3) Sensitivity to the Absence of the Translation Block:
Following the architecture used in general I2I translation tasks, we use a translation block in our translator architecture. Therefore, we also conducted experiments to study the sensitivity of our method to the absence of the translation block. From the visualization results in Figure 4.7, we find that there is a noticeable quality degradation in visuals without the block. Both genuine face and attack images become blurred. However, as in Table 4.14, there is no severe decline in the numerical performance. Since the translation is only an auxiliary task to improve the final attack detection accuracy, our experimental results indicate the potential for architecture pruning to enhance efficiency further if necessary.

4) Compatibility of Illumination Normalization Module with Existing Methods:
Since the proposed IN module works as a preprocessing procedure and can be easily added to other multi-modality face PAD methods, we finally test the compatibility of IN module with two baseline methods. From Table 4.15, we find that the PLGF-based illumination normalization also improves the performance of the MC-PixBiS



(A) (B) (C) (D) (E) (F)

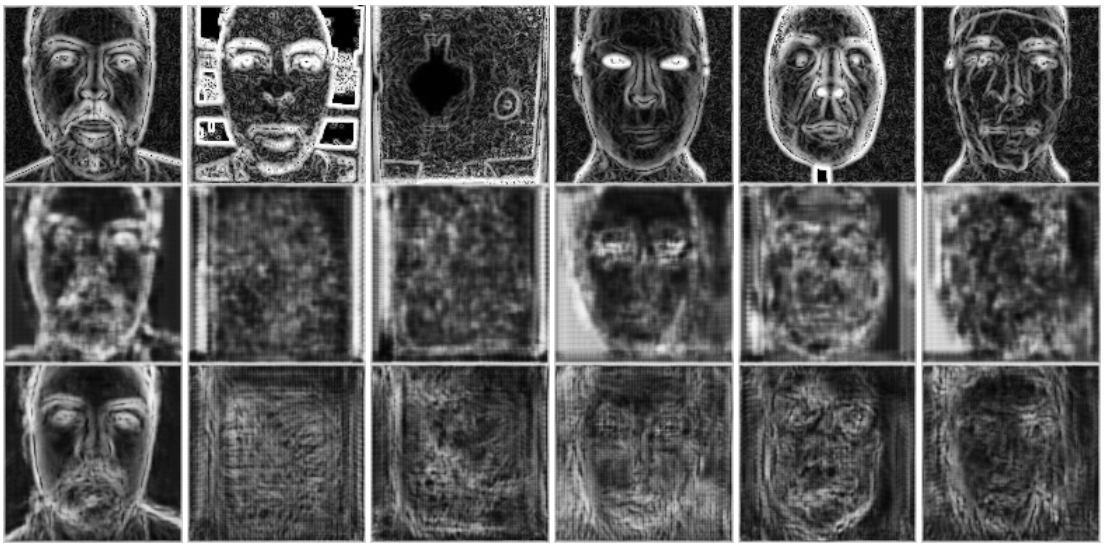
FIGURE 4.6: The figure shows the visualization of images under WMCA (T-I) setting. Column (A) are images of a same genuine face sample, and columns (B)-(F) are images of different types of attacks. From top to bottom row are images of ground-truth thermal, ground-truth NIR, NIR that translated from thermal, ground-truth IN-NIR and IN-NIR that translated from thermal images. Noted that samples of glasses attack and fake head attack are not visualized here because no samples are in authorized list due to the privacy issue.

method, especially under CASIA-SURF (D-I) evaluation setting. While there is no obvious improvement for MM-CDCN, we believe it is because the CDC used in MM-CDCN itself possesses kernel-wised normalization functionality.

4.4.6 Discussions

1) Cross-Domain Evaluation Performance:

According to the experimental results, our method achieves good performance in grand-test and cross-illumination evaluation. However, the performance of cross-dataset evaluation is not satisfactory. It is because different camera sensors and



(A) (B) (C) (D) (E) (F)

FIGURE 4.7: The figure shows the visualization of images under WMCA (T-I) setting. Column (A) are images of a same genuine face sample from WMCA, and columns (B)-(F) are images of different attacks. From top to bottom row are images of ground-truth IN-NIR, translated IN-NIR without and with the translation block.

TABLE 4.14: Experimental Results with/without Translation Blocks

Modality Setting	Variants	Metrics(%)				
		AUC ↑	TDR@FDR=1% ↑	APCER ↓	BPCER ↓	ACER ↓
WMCA(T-I)	w/o TB [112]	99.94	99.52	0.55	0.68	0.61
	w/ TB [112]	**99.99**	**99.71**	**0.45**	**0.02**	**0.23**
CASIA-SURF(D-I)	w/o TB [112]	99.90	98.25	1.75	**0.76**	1.25
	w/ TB [112]	**99.97**	**99.66**	**0.20**	1.73	**0.96**

TABLE 4.15: Experimental Results with Illumination Normalization Module

Modality Setting	Variants	Metrics(%)				
		AUC ↑	TDR@FDR=1% ↑	APCER ↓	BPCER ↓	ACER ↓
WMCA(T-I)	MC-PixBiS [9]	99.12	96.34	3.84	**0.71**	2.28
	MC-PixBiS [9] + IN	**99.22**	**97.69**	**2.33**	0.94	**1.63**
CASIA-SURF(D-I)	MC-PixBiS [9]	99.30	87.84	7.63	1.61	4.62
	MC-PixBiS [9] + IN	**99.78**	**97.31**	**2.02**	**1.60**	**1.81**
WMCA(T-I)	MM-CDCN [102]	**99.20**	96.05	3.94	**1.08**	2.51
	MM-CDCN [102] + IN	99.16	**96.80**	**2.58**	1.55	**2.06**
CASIA-SURF(D-I)	MM-CDCN [102]	99.73	95.74	**2.38**	1.78	2.08
	MM-CDCN [102] + IN	**99.80**	**96.63**	**2.38**	**1.71**	**2.04**

data preprocessing schemes cause large domain shift between the samples of different datasets, which severely degrades the accuracy of face PAD models. In pratical applications, the specifications of sensors and data preprocessing schemes should be unified to avoid the performance loss. Besides, the performance for the detection of unseen attacks such as disguising glasses attacks could be further improved.

TABLE 4.16: Grand-Test Experimental Results on WMCA (Score Fusion)

Method	Metrics(%)				
	AUC	TDR@FDR=1%	APCER	BPCER	ACER
	↑	↑	↓	↓	↓
T-V [112]	99.919	98.99	0.86	2.05	1.46
T-I [112]	99.989	99.71	0.45	**0.02**	**0.23**
Score Fusion [112]	**99.995**	**99.84**	**0.30**	0.17	**0.23**

2) Extension to K-Modality Settings:

Although our method is proposed to address the face PAD problem under bi-modality scenarios, we attempted to extend it to K(K>2) modality scenarios by performing score fusion with different bi-modality models. We did some experiments on WMCA and CASIA-SURF datasets in 3 modality scenarios, and the results are shown in Table 4.16 and Table 4.17, respectively. The score fusion is realized by performing weighted sum with the scores of two bi-modality models, the weights for the more and less accurate models are set as 0.8 and 0.2, respectively. According to our experimental results, the AUC, TDR@FDR=1%, and APCER can be further improved with the score fusion of different bi-modality models. However, naively extending the proposed method with multiple model score fusion will multiply the model size and the computational cost.

3) Data Augmentations:

The good performance of deep learning-based face PAD methods relies on sufficient and diverse training data. However, it is usually hard to collect enough training data to cover the test domain in practical applications. Data augmentation techniques such as brightness, contrast, saturation, and hue adjustment have been used for the single VIS-modality face PAD method to improve the accuracy and generalization performance. But most of the data augmentation techniques for face PAD methods are specifically designed for VIS images. They can not be directly applied to other modalities like depth maps and thermal images. The exploration of data augmentation techniques for multi-modality data is a promising direction to further improve the generalization performance of multi-modality face PAD methods.

4) Different Evaluation Metrics:

In our experiments, we evaluated the performance with different metrics such as AUC, TDR@FDR=1%, APCER, BPCER, and ACER. By visualizing the ROC curves as in Figure 4.8, we found that both baseline methods and our method achieve close to perfect TDRs at the thresholds of higher FDRs, but our method

TABLE 4.17: Grand-Test Experimental Results on CASIA-SURF (Score Fusion)

Method	Metrics(%)				
	AUC	TDR@FDR=1%	APCER	BPCER	ACER
	↑	↑	↓	↓	↓
D-V [112]	99.965	99.45	0.45	**1.02**	**0.74**
D-I [112]	99.968	99.66	0.20	1.73	0.96
Score Fusion [112]	**99.988**	**99.91**	**0.05**	1.79	0.92

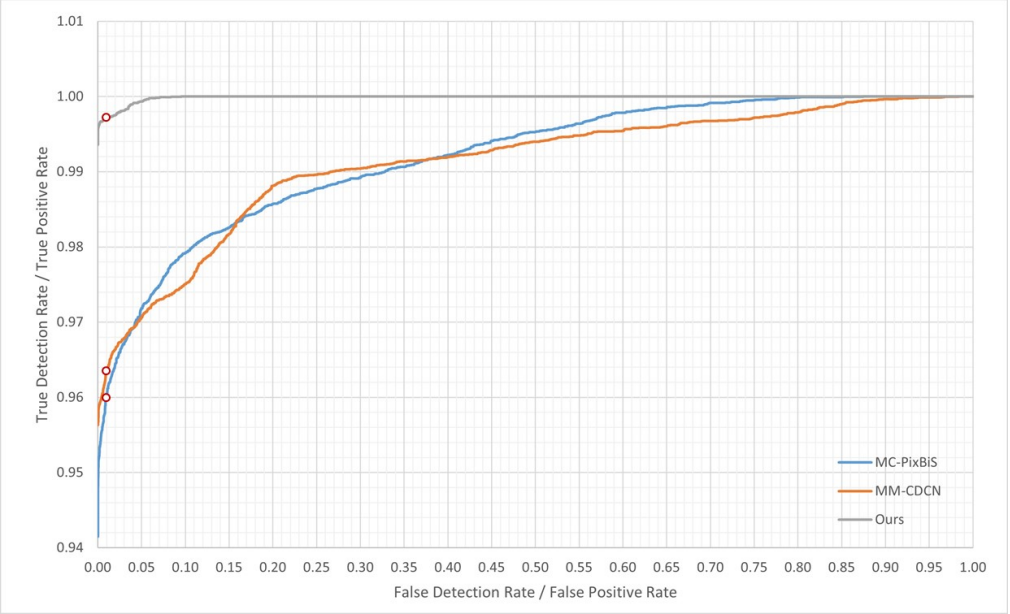


FIGURE 4.8: Visualization of Receiver Operating Characteristic (ROC) curves for WMCA(T-I) grand-test evaluation. The horizontal axis represents the False Detection Rate (FDR) and the vertical axis represents the True Detection Rate (TDR).

shows significant superiority with lower FDRs. AUC measures the overall performance at the thresholds of different FDRs. While metrics TDR@FDR=1%, APCER, BPCER, and ACER measure the performance at fixed thresholds of lower FDR only. Therefore, the performance improvement of our method in terms of AUC is not as large as other metrics.

5) Comparison Between Different Modalities:

To verify the effectiveness of our proposed method, we did experiments under different bi-modality settings of WMCA and CASIA-SURF datasets. On the WMCA dataset, our method with T-I modality outperforms V-I and T-V modality. On the CASIA-SURF dataset, our method with D-V and D-I modality performs better than the V-I modality, and the difference between the D-V modality and D-I modality is not significant.

4.5 Chapter Summary

In this chapter, we tackle the generalization problems in face PAD under the bi-modality scenarios and propose a method that better generalizes to unseen attack and illumination variations. Different from existing methods, we explicitly establish the connection between face images of different modalities via asymmetric modality translation, which can successfully transform genuine face images from the source to the target modality but fails for attack ones. The discrepancy of modality translation between genuine faces and attack samples is used as a compelling clue for discriminating various spoofing faces from genuine faces. Besides, an illumination normalization (IN) module based on PLGF descriptor is used to alleviate the interference of the illumination variations on sensitive modalities.

To verify the effectiveness of our proposed method, we did extensive experiments on several public datasets. The experimental results show that our method applies to different modality settings and can effectively detect various seen and unseen attacks under varying illumination conditions. However, the performance of the method in cross-dataset evaluation is not satisfactory. We find that the cross-dataset generalization problem on multi-modality face PAD research may not be easier than single VIS-modality ones.

Chapter 5

One-Class Knowledge Distillation for Face PAD

5.1 Introduction

Existing face PAD methods have achieved good performance in intra-domain testing, where the testing data is from the same distribution as training data. However, when testing the face PAD models in a new target domain, the performance will degrade severely since the testing data is from unseen distributions which are different from the training data. This problem is also known as the distribution shift or domain shift problem, which originates from various factors such as the change of capturing devices, mediums of attacks, and illumination [3, 4].

Since the domain shift seriously affects the reliability of face PAD models, domain adaptation techniques have recently been used to address the cross-domain problems in face PAD. Domain adaptation techniques improve the cross-domain performance by utilizing the target domain data. However, it is difficult and expensive to collect and annotate sufficient data samples in the target domain for the adaptation. Moreover, collecting attack samples requires facial forgeries. The production of facial forgeries is complicated, and there is no guarantee that the collected attack samples are the same as those launched by attackers. Compared to attack samples, genuine face samples are much easier and cheaper to collect. Therefore, we expect to improve the cross-domain performance of the face PAD model by only using a few genuine face samples collected in the target domain,

which is named one-class domain adaptation (OCDA) [95]. The straightforward approach for OCDA is to train a face PAD model with the source domain data and fine-tune the pre-trained model with the target domain data. However, such a naive fine-tuning approach cannot provide good performance due to the one-class characteristic of the target domain data. Recently, Qin *et al.* [95] propose a meta-learning method, which incorporates a meta loss function search (MLS) strategy to search for better loss functions and help the meta-learner deal with the OCDA tasks. Mohammadi *et al.* [94] propose a method to tackle the OCDA tasks with domain guided pruning. In this method, the generalization ability of different filters in the pre-trained face PAD model is estimated with the feature divergence between genuine face samples from the source and the target domain. The filters with poor generalization ability are pruned. Besides, Fatemifar *et al.* [123] propose to develop client-specific face PAD models with some genuine face samples collected in the target client domain. Pre-trained deep neural networks are used as feature extractors to build face PAD models with conventional one-class classifiers. Recent works show that it is promising to facilitate the training of face PAD models with some genuine face samples collected in the target domain. However, the performance of existing methods needs to be further improved, especially in scenarios where mixture factors cause a large distribution shift between the source and the target data domain. How to leverage the genuine face samples to improve the target domain performance of face PAD effectively is still a crucial problem to be studied.

In this chapter, we tackle the OCDA problem in face PAD with teacher-student learning. Different from previous knowledge distillation based face PAD methods [86, 90], our method tackles the OCDA problem of face PAD with a teacher-student network by drawing lessons from the anomaly detection task [124]. In our framework, a teacher network is trained with the source domain data to provide discriminative feature representations for face PAD. Inspired by anomaly detection [124], the student network is trained with only genuine face samples of the target domain to generate similar representations to the teacher's outputs. As such, the student network is "stubborn" and only learns similar representations for the genuine face images. Therefore, the genuine face representations of the teacher and student networks are more similar than the spoof ones. Finally, we use the similarity score to discriminate attacks from the genuine ones, as illustrated in Figure 5.1. Moreover, recent literature [93, 123] points out that different target

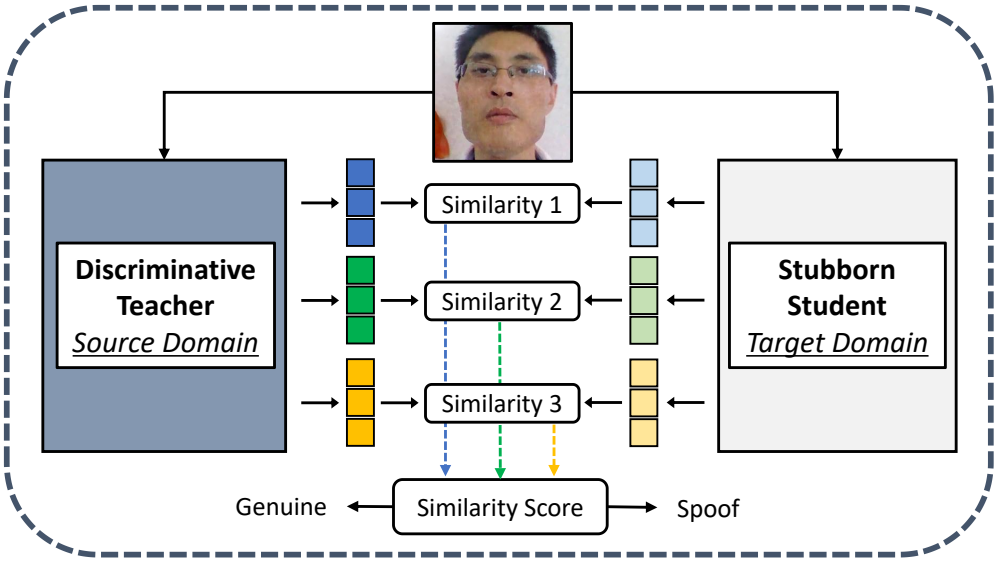


FIGURE 5.1: The figure illustrates the framework under the general setting. It contains a teacher network trained with the source domain data to provide discriminative features, and a student network trained with the target domain genuine face data to generate similar features to the teacher's descriptions. In the test phase, the face images will be fed into both the DT and SS netorks for feature extraction and the similarity between features of the two networks will be used as the inference score.

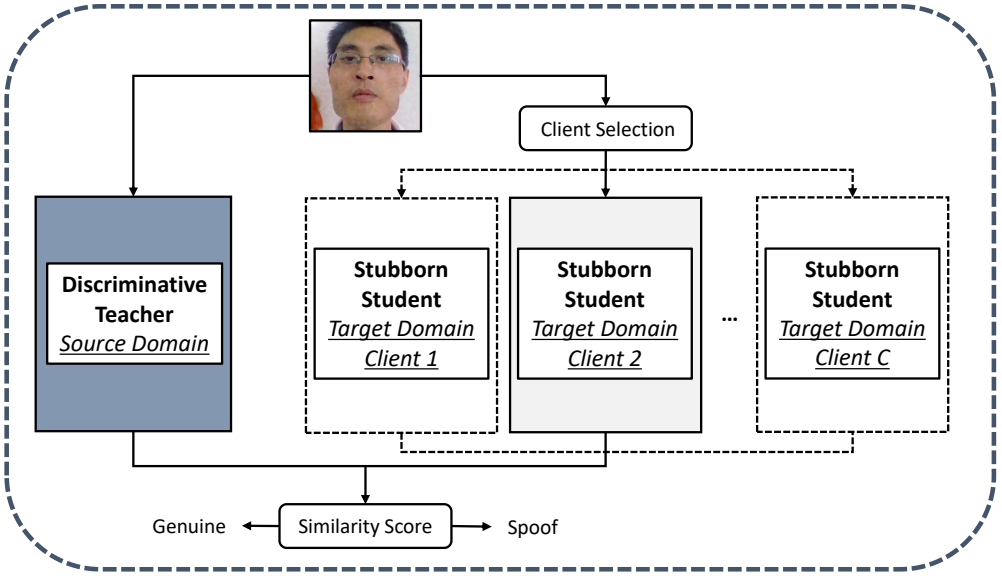


FIGURE 5.2: The figure illustrates the framework under the client-specific setting. After the client-specific one-class domain adaptation, the framework contains a teacher network and a set of N student networks. Each student network serves for one specific target client. In the test phase, the face images will be fed into the DT network and the corresponding SS network for client-specific inference.

clients could be reckoned as different client-specific domains. Therefore, we also consider the OCDA problem under a client-specific setting. Client-specific one-class domain adaptation (CS-OCDA) aims to develop a specific model for each target client with a few of its own genuine face samples. The proposed method is flexible to be extended for the CS-OCDA by training a "stubborn" student model for each client. Although the use of the student networks multiplies the number of parameters as the number of clients increases, we apply a sparse learning strategy [125] during the optimization to shrink the size of student networks and alleviate the storage problem. Our main contributions in this work [1] are summarized as below:

- We introduce a one-class knowledge distillation framework to address the cross-domain problem in face PAD, which improves the target domain performance by utilizing only a few genuine face samples in the target domain.
- We devise two new protocols on public benchmark datasets for the performance evaluation of face PAD methods under the general and client-specific one-class domain adaptation.
- We conduct extensive experiments to verify the effectiveness of our proposed method. The experimental results show that our method outperforms baseline methods under one-class domain adaptation settings and even state-of-the-art methods with unsupervised domain adaptation.

5.2 Methodology

5.2.1 Problem Formulation

Existing face PAD models trained with the source domain data can not generalize well to the target domain data. Although domain adaptation techniques could improve the cross-domain performance of face PAD with the help of the data collected in the target domain, the data collection is expensive and complicated, especially for attack ones. Therefore, we expect to address the cross-domain problem in face PAD with one-class domain adaptation (OCDA). In addition to source domain

[1] The work in this chapter has been published in [126].

training data, we aim to improve the cross-domain performance of face PAD by only using a few genuine face samples collected in the target domain. We formulate the OCDA problem of face PAD under both the general and client-specific settings. For general one-class domain adaptation, different target clients are reckoned as one general domain, and the objective is to develop a general model for all clients. For client-specific one-class domain adaptation, different target clients are reckoned as different client-specific domains. The objective is to develop a set of client-specific models. Each model serves one specific target client.

5.2.2 Framework of the Proposed Method

Inspired by the advances in anomaly detection [124], we introduce a teacher-student framework named one-class knowledge distillation framework to address the one-class domain adaption problem of face PAD. The method applies to both the general and client-specific OCDA settings. A teacher network θ_{DT} is trained with the genuine and attack samples from the source domain D_{src} to provide discriminative feature representations for face PAD. For the general one-class domain adaptation setting, a student network θ_{SS} is trained with only genuine data from the general target domain D_{tgt} to generate similar representations to the teacher's outputs. As such, the genuine face representations of the teacher and student networks are more similar than the spoof ones. In the testing phase, we use the similarity score to detect attacks, as illustrated in Figure 5.1. For client-specific one-class domain adaptation setting, a set of stubborn student networks θ^c_{SS} are trained with genuine data from multiple client-specific target domains D^c_{tgt}, each student network serves for a specific target client, as illustrated in Figure 5.2. Although the use of the student networks multiplies the number of parameters, a sparse learning strategy [125] is adopted during the training of the student networks to alleviate the expansion of the model size. In the following contents of this chapter, we'll introduce the design details and training strategies of the teacher and student networks.

5.2.3 Discriminative Teacher Stream

The function of the Discriminative Teacher (DT) is to provide discriminative feature representations for the face PAD task. Benefit by the strong representational

Algorithm 1: Training of the Discriminative Teacher Stream

Input: Source domain training data D_{src}, learning rate α_1, maximum training iteration K_1, and batch size N_1.

Output: DT parameters θ_{DT}.

1: Initialize the DT parameters as θ_{DT} and the FCB parameters as θ_{FCB}.
2: **for** $k = 1$ **to** K_1 **do**
3: Sample N_1 samples x_i with the labels y_i from D_{src}.
4: Extract multi-level features f_i^1, f_i^2, and f_i^3 by encoding x_i with θ_{DT}.
5: Predict the pixel map d_i by processing f_i^1, f_i^2, and f_i^3 with θ_{FCB}.
6: Compute $\mathcal{L}_{DT}$ with d_i and y_i as in Eq. (1).
7: Update θ_{DT} and θ_{FCB} with $\mathcal{L}_{DT}$, α_1.
8: **end for**
9: **return** θ_{DT}

ability of deep learning, convolutional neural networks are widely used as the feature extractors for face PAD. We use the feature extractor of the depth regression network proposed in [60] as the backbone of our DT. A series of convolutional blocks are used to encode image x_i into 3 level feature representations f_i^1, f_i^2, and f_i^3. In addition, a final convolutional block takes the multi-level features to estimate the pixel map d_i. To avoid the inconvenience of the pseudo depth map generation, we use binary maps y_i with a 0/1 value as the target of the pixel map, similar to [66]. Binary cross-entropy is used for loss calculation with d_i and the target y_i at the pixel-level, which is represented as

$$\begin{aligned} \mathcal{L}_{DT} &= \frac{1}{N_1}\sum_{i=1}^{N_1} -(y_i \log(d_i) + (1-y_i)\log(1-d_i)), \\ y_i &= \begin{cases} 0, & genuine, \\ 1, & attack. \end{cases} \end{aligned} \tag{5.1}$$

We use θ_{DT} and θ_{FCB} to denote the parameters of the feature extractor and the final convolutional block. The DT is trained with genuine and attack samples from D_{src}. The optimization problem is represented as,

$$\underset{\theta_{DT},\theta_{FCB}}{\arg\min} \quad E_{x,y\sim D_{src}}\mathcal{L}_{DT}(x, y|\theta_{DT}, \theta_{FCB}). \tag{5.2}$$

Figure 5.3 illustrates the framework of the DT stream. The Algorithm 1 describes the details about the training of the DT. Since the expected function of the teacher

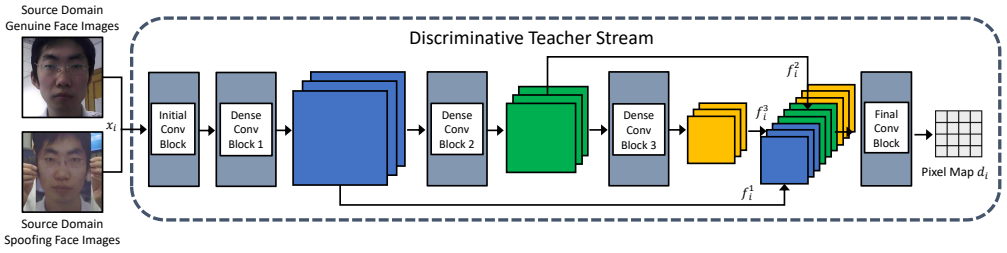


FIGURE 5.3: The figure illustrates the network training of the Discriminative Teacher network with the source domain data. The face image x_i will be fed into several convolutional blocks to extract features f_i^1, f_i^2 and f_i^3. The multi-level features are fed into the final convolutional block for pixel map d_i estimation.

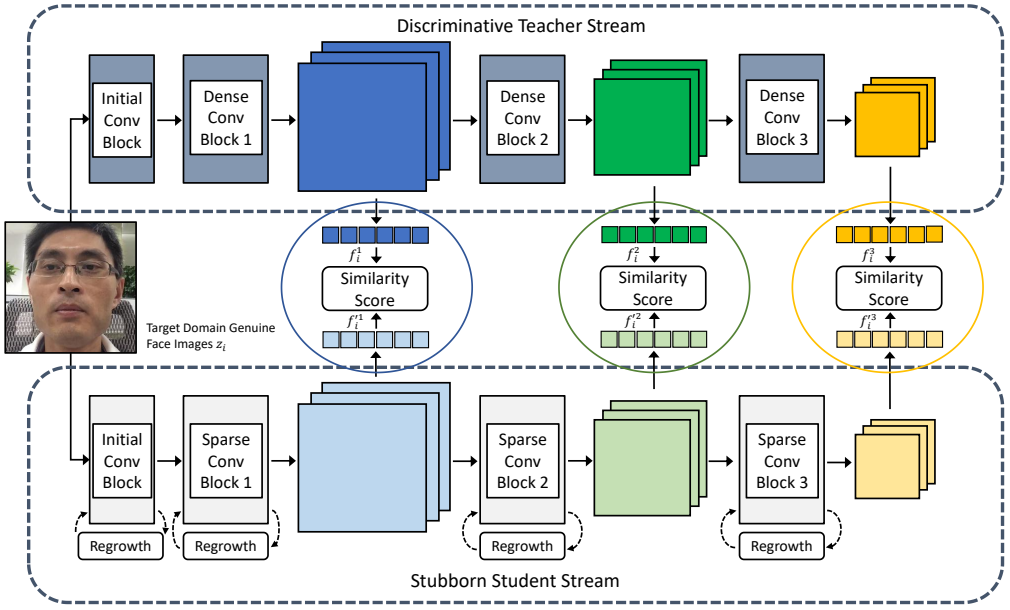


FIGURE 5.4: The figure illustrates the training of the Stubborn Student network. The target domain genuine face images z_i will be fed into both the teacher and student networks for feature extraction. The multi-level similarities between the DT and SS networks are computed to guide the optimization of the SS network. During the training, a sparse training strategy is used to reduce the parameter density of the SS network.

stream is to provide discriminative features for face PAD only, the final convolutional block is abandoned after the DT network training.

5.2.4 Stubborn Student Stream

The learning objective of the Stubborn Student (SS) is to generate similar genuine face representations to the teacher's descriptions. We expect the student network is inflexible so that the representations of the student network are similar to the

teacher's representations only for the genuine face images. During the training, genuine face images z_i from the target domain D_{tgt} are fed into both the DT and SS for feature extraction. f_i^1 f_i^2, and f_i^3 denote the representations of DT, while $f_i^{'1}$, $f_i^{'2}$ and $f_i^{'3}$ denote the representations of SS. The similarity between the DT and the SS representations is measured by the cosine distance similar to [90, 112],

$$\mathcal{S}(f, f^{'}) = 1 - \frac{\langle f, f^{'} \rangle}{|f||f^{'}|}, \tag{5.3}$$

where f, f' denote feature vectors, and $\langle \cdot \rangle$ denotes the inner product. Similarities are computed at 3 feature levels, and the weighted sum of them constitutes the $\mathcal{L}_{SS}$ to direct the training of the SS network. The representation of $\mathcal{L}_{SS}$ is,

$$\mathcal{L}_{SS} = \frac{1}{N_2} \sum_{i=1}^{N_2} \lambda_1 \mathcal{S}(f_i^1, f_i^{'1}) + \lambda_2 \mathcal{S}(f_i^2, f_i^{'2}) + \lambda_3 \mathcal{S}(f_i^3, f_i^{'3}). \tag{5.4}$$

We denote the parameters of DT and SS as θ_{DT} and θ_{SS}. The optimization problem for SS is represented as,

$$\arg\min_{\theta_{SS}} \quad E_{z \sim D_{tgt}} \mathcal{L}_{SS}(z|\theta_{DT}, \theta_{SS}). \tag{5.5}$$

The use of the SS networks multiplies the number of parameters in the face PAD model. The problem is more critical under the client-specific one-class domain adaptation setting since a set of SS networks needs to be stored together with the DT network. To relieve the pressure about the expansion of the model size, sparse SS networks are trained in our method. The architecture of the SS network is similar to the DT network, and the only difference is that the convolution kernels of the SS network are sparse. For each convolution layer, only $s\%$ number of parameters are non-zero.

A sparse training strategy [125] is adopted during the optimization of the SS networks. We denote $s\%$ as the desired density of the SS network, and $(1-s\%)$ number of parameters are set as zero and inactive for the update. We use $v_{m,n}$ to denote the initial indicator of the n-th parameter in the m-th convolution layer and τ_m to denote the threshold. The inactive parameters are determined by the magnitude

Algorithm 2: Training of the Stubborn Student Stream

Input: Target domain training data D_{tgt}, DT parameters θ_{DT}, SS density $s\%$, learning rate α_2, maximum training iterations K_2, batch size N_2, regrowth period T, and initial regrowth rate $r\%$.

Output: SS parameters θ_{SS}

Initialize the SS parameters as θ_{SS}.
Calculate the number of parameters in each convolutional layer. l_m denotes the number for the m-th layer.
Determine the indices A_m for the $l_m \cdot s\%$ active parameters as in Eq. (6) and B_m for the $l_m \cdot (1 - s\%)$ inactive parameters as in Eq. (7).
Deactivate the parameters $\omega_{m,n}$ for $n \in B_m$ by setting them 0.
for $k = 1$ **to** K_2 **do**
    Sample N_2 samples z_i from D_{tgt}.
    Extract features f_i^1, f_i^2, f_i^3 by encoding z_i with θ_{DT}, and $f_i'^1$, $f_i'^2$, $f_i'^3$ by encoding z_i with θ_{SS}.
    Compute $\mathcal{L}_{SS}$ with f_i^1, f_i^2, f_i^3, $f_i'^1$, $f_i'^2$, $f_i'^3$ as Eq. (4).
    Update the active parameters of θ_{SS} with $\mathcal{L}_{SS}$, α_2, A_m.
    Adjust the regrowth rate $r\%$ with cosine decay.
    if $k \; mod \; T \; = \; 0$ **then**
        Calculate the number of the active parameters $|A_m|$.
        Determine the indices P_m for $|A_m| \cdot r\%$ parameters to be pruned according to Eq. (8).
        Update the indices of the active and inactive parameters by $A_m \leftarrow A_m - P_m$ and $B_m \leftarrow B_m \cup P_m$.
        Determine the indices G_m for $|A_m| \cdot r\%$ parameters to be grown according to Eq. (9) - Eq. (12).
        Update the indices of the active and inactive parameters by $A_m \leftarrow A_m \cup G_m$ and $B_m \leftarrow B_m - G_m$.
    end if
end for
return θ_{SS}

of the indicators. The sets of the active and inactive parameter indexes for the m-th convolution layer are presented in Eq. (5.6) and Eq. (5.7), respectively.

$$A_m = \{n||v_{m,n}| \geq \tau_m\}, \tag{5.6}$$

$$B_m = \{n||v_{m,n}| < \tau_m\}. \tag{5.7}$$

During the training, the sets of the active and inactive parameters are periodically adjusted with a regrowth mechanism [125] to optimize the SS network training. The

parameter regrowth mechanism consists of the pruning and growing operations. The pruning refers to that the active parameter is set inactive, and the growing refers to the inactive parameter is set active. The operations are applied to the convolution layer of the SS network with a period T. After T iterations, the $r\%$ active parameters of each convolution layer will be pruned. The pruning scheme is based on the relative importance of the parameter in the layer, and the importance is measured by the magnitude of the parameter value $|\omega_{m,n}|$. The parameter will be pruned if its magnitude is smaller than the pruning threshold τ_m^p, and the indexes set of parameters to be pruned represented as

$$P_m = \{n | n \in A_m, |\omega_{m,n}| < \tau_m^p\}. \tag{5.8}$$

After the pruning, the same number of the inactive parameters will be grown according to their ability to reduce the loss $\mathcal{L}_{SS}$. Following [125], the estimation of the ability $\mu_{m,n}$ is based on the momentum values $p_{m,n}$ and $q_{m,n}$ as represented in Eq.(5.9) and Eq.(5.10).

$$p_{m,n} = \beta_1 p'_{m,n} + (1 - \beta_1)\frac{\partial \mathcal{L}_{SS}}{\partial \omega_{m,n}}, \tag{5.9}$$

$$q_{m,n} = \beta_2 q'_{m,n} + (1 - \beta_2)(\frac{\partial \mathcal{L}_{SS}}{\partial \omega_{m,n}})^2, \tag{5.10}$$

The $p_{m,n}$ and $q_{m,n}$ are the first and the second order momentum of the current iteration. $p'_{m,n}$ and $q'_{m,n}$ are the first and the second order momentum of the last iteration. β_1 and β_2 are smoothing factors. The value of the momentum $p_{m,n}$ and $q_{m,n}$ can be easily computed with the optimizer class implemented in PyTorch [127]. The computation of the $\mu_{m,n}$ is represented as

$$\mu_{m,n} = \frac{p_{m,n}}{\sqrt{q_{m,n}} + \epsilon}. \tag{5.11}$$

ϵ is a small constant 10^{-8} to avoid the denominator being equal to 0. By comparing the $|\mu_{m,n}|$ with the threshold τ_m^g, the indexes set of parameters to be grown is represented as,

$$G_m = \{n | n \in B_m, |\mu_{m,n}| \geq \tau_m^g\}. \tag{5.12}$$

Figure 5.4 illustrates the framework of the SS stream. The Algorithm 2 describes the details about the training of the SS network.

5.2.5 Inference at the Test Phase

After the student network training, both the teacher network DT and student networks SS are used to compose the inference model. The illustrations for general and client-specific tasks are shown in Figure 5.1 and Figure 5.2, respectively. The image sample I_t is fed into both the DT and SS networks to generate feature representations f_t^1, f_t^2, f_t^3, and $f_t'^1$, $f_t'^2$, $f_t'^3$, respectively. The inference score ξ_t is computed with the three similarities between features of the two networks, which is represented as,

$$\xi_t = \lambda_1 \mathcal{S}(f_t^1, f_t'^1) + \lambda_2 \mathcal{S}(f_t^2, f_t'^2) + \lambda_3 \mathcal{S}(f_t^3, f_t'^3). \tag{5.13}$$

As shown in Eq. (5.14), if the score ξ_t is smaller than the threshold δ, the image I_t is determined as a genuine face. Otherwise, the image I_t is determined as an attack.

$$I_t = \begin{cases} genuine, & \xi_t < \delta, \\ attack, & \xi_t \geq \delta. \end{cases} \tag{5.14}$$

5.3 Experimental Setup

5.3.1 Dataset Information

To verify the effectiveness of our proposed method, we devise two protocols for the performance evaluation of face PAD models under the general and client-specific one-class domain adaptation settings. We conduct experiments on 5 datasets commonly used for cross-domain face PAD evaluation, including CASIA-FASD [104], MSU-MFSD [19], IDIAP REPLAY-ATTACK [21], NTU ROSE-YOUTU [51], and OULU-NPU [3] datasets.

5.3.2 Evaluation Protocols

1) General One-Class Domain Adaptation:
To evaluate the performance of face PAD under the general one-class domain adaptation setting, we devise a protocol named CIMN One-Class Domain Adaptation (CIMN-OCDA) with CASIA-FASD (C), IDIAP REPLAY-ATTACK (I), MSU-MFSD (M), and NTU ROSE-YOUTU (N) datasets, which are commonly used for the evaluation of face PAD methods under the unsupervised domain adaptation settings [51, 89, 92]. For the CIMN-OCDA protocol, each dataset is considered as a data domain. To simulate the cross-domain scenarios, each domain could be set as the source domain and paired with the others as the target domains to form 3 one-class domain adaptation tasks. We denote the tasks by the abbreviations of the datasets. For example, C-I denotes the task with the CASIA-FASD as the source domain and the IDIAP REPLAY-ATTACK as the target domain. For each task, all data of the source domain *train* set and the genuine face data of the target domain *train* set are used for model training, while all the data of the target domain *test* set are used for performance evaluation. In addition to the cross-dataset experiments on the CIMN-OCDA protocol, we also conduct experiments on the OULU One-Class Adaptation Structure A (OULU-OCA-SA) protocol proposed in [95] for the comparison with similar methods. To verify the effectiveness under the cross-illumination settings, we devise a protocol with WMCA dataset named WMCA Cross-Illumiantion One-Class Domain Adaptation (WMCA-CI-OCDA). For each sub-protocol of WMCA-CI-OCDA, we leave image samples under one illumination as the target domain and others as the source domain.

2) Client-Specific One-Class Domain Adaptation:
For the client-specific one-class domain adaptation task, the target domain is client-specific, and the objective is to develop a specific model with better performance for each target client with a few genuine face images. To evaluate the performance under the client-specific one-class domain adaptation setting, we devise a protocol named CIM-N Client-Specific One-Class Domain Adaptation (CIM-N-CS-OCDA). Since the performance evaluation requires a large amount of data for each target client, we employ the NTU ROSE-YOUTU dataset and sample 10 clients to simulate the client-specific target domains. Each target client domain contains 50 genuine face videos and 110 attack videos. We uniformly divide 25 genuine face videos together with all 110 attack videos as testing data. Then we sample 1 frame

from each of the remaining 25 genuine face videos as the target domain training data. We set the CASIA-FASD (C), IDIAP-REPLAY ATTACK (I), and MSU-MFSD (M) datasets as the source domains to form 3 sub-protocols. For short, we denote the sub-protocols as C-N-CS, I-N-CS, and M-N-CS in the following content. Each sub-protocol contains 10 client-specific one-class domain adaptation tasks for performance evaluation.

5.3.3 Evaluation Metrics

The Half Total Error Rate (HTER) and the Area Under Receiver Operating Characteristic Curves (AUC) are the most commonly used metrics for evaluating face PAD methods under cross-domain settings. The HTER is the average of the False Accept Rate (FAR) and the False Reject Rate (FRR), measuring the error rate of the face PAD model at a fixed threshold. As a complement, the AUC is a comprehensive metric that measures the overall performance over different thresholds. Besides, the Average Classification Error Rate (ACER) is the average of the Attack Presentation Classification Error Rate (APCER) and the Bonafide Presentation Classification Error Rate (BPCER). For experiments on the OULU-OCA-SA protocol, we use the ACER and AUC as metrics for a fair comparison with existing methods.

5.3.4 Baseline Methods

To verify the effectiveness of our proposed method, we implement 4 face PAD methods as the baselines under the one-class domain adaptation settings. Besides, we evaluate our method on the OULU-OCA-SA protocol and compare the performance with existing methods addressing face PAD under similar scenarios. Even though our method does not use any target domain attack samples for model training, we compare our method with unsupervised domain adaptation methods, which use unlabelled genuine face and attack samples for model training.

1) DT:
As introduced in Chapter 5.2.3, the depth regression network [60] is employed as the backbone of DT and trained with the source domain data. Following [60], we compute the average value of the pixel map d_i as the inference score and set the

result as the baseline performance of face PAD without using target domain data for adaptation.

2) DT + Fine-Tune:
This method is a naive extension to the DT. The only difference is that we use the genuine face samples of the target domain *train* set to fine-tune the model after pre-training the DT with the source domain data.

3) DT + OCSVM:
Recently, deep neural networks pre-trained on the image classification datasets have been used as feature extractors to develop face PAD models with only some genuine face samples collected in the target domain. Following [123], we implement a baseline based on one-class support vector machine (OCSVM) [41]. We use the pretrained DT with the source domain data as the feature extractor and train an OCSVM classifier for face PAD with the target domain genuine face samples.

4) DT + GMM:
Gaussian mixture model (GMM) [42] is a parametric probability density function, which is commonly used for one-class classification problems. Following [123], we also implement a method that trains a GMM-based face PAD model with the DT features as our baseline.

5) OCA-FAS:
OCA-FAS [95] is a recent method for face PAD under the one-class domain adaptation setting. Since it is the most relevant work to ours in the literature, we also conduct experiments on the OULU-OCA-SA protocol to compare with it.

6) Others:
In addition to face PAD methods with one-class domain adaptation, we also compare our method with face PAD methods using unsupervised domain adaptation such as KSA [51], ADA [89], UDA [92], USDAN-Un [91], etc.

5.3.5 Implementation Details

We apply the same scheme for data pre-processing on all 5 datasets as follows. We uniformly sample 50 image frames from each video clip. The dlib library [2] is used

[2]https://github.com/davisking/dlib

for face detection and alignment. After that, the cropped face regions are further resized to 128×128. After the data pre-processing, we get 30K, 58.4K, 13.7K, 161.95K, and 246.75K images for CASIA-FASD, IDIAP REPLAY-ATTACK, MSU-MFSD, NTU ROSE-YOUTU, and OULU-NPU datasets. The training of the DT and the SS networks are both optimized with the Adam optimizer [128]. For the training of the DT network, the mini-batch size and learning rate are set as 30 and 10^{-4}, respectively. We train the DT network for 8400 iterations. For the training of the SS network, the mini-batch size and learning rate are set as 25 and 10^{-4}, respectively. The weights λ_1, λ_2, λ_3 in Eq. (5.4) and Eq. (5.13) are all set as 0.33. The regrowth period T is set as 60. The initial regrowth rate $r\%$ is set as 50% and 20% for the experiments of 10% and 1% SS density and adjusted with cosine decay. We train the SS network for 1500 iterations. The proposed and baseline methods are implemented based on PyTorch [127] version 1.7.0.

5.4 Results and Analysis

5.4.1 General One-Class Domain Adaptation Experiments

1) Results on the CIMN-OCDA Protocol:
To verify the effectiveness of our proposed method under the general one-class domain adaptation setting, we firstly conduct experiments on the CIMN-OCDA protocol. For fair comparisons, all the baseline methods and our method share the same DT model as the feature extractor. The SS model density is set as 10%. We evaluate the cross-dataset performance of face PAD methods under two different experimental settings, which are referred to as the ideal and challenging experimental settings, respectively. For the ideal experimental setting, the HTER performance is directly computed on the test set of the target dataset at the optimal threshold. The experimental results of the ideal setting are shown in Table 5.1. For the challenging experimental setting, the HTER performance is computed at the threshold that is pre-determined on a validation set where the FRR=10%. For experiments that use dataset I as the target dataset, we use the genuine face samples of the development set as the validation set. Since there is no development set or validation set in datasets C, M, and N, for experiments that use C, M, or N as the target dataset, we divide 20% genuine face samples from the train set of the

TABLE 5.1: Performance Comparison with the One-Class Domain Adaptation Methods on the CIMN-OCDA Protocol (ideal experimental setting)

Method	HTER (%) ↓												
	C-I	C-M	C-N	I-C	I-M	I-N	M-C	M-I	M-N	N-C	N-I	N-M	Average
DT [60]	36.3	**14.1**	28.4	45.1	35.9	43.2	35.0	23.4	38.2	28.9	29.0	21.7	31.6
DT + Fine-tune	41.1	23.2	35.7	49.4	**18.6**	35.7	29.5	36.0	36.2	**19.7**	27.7	19.4	31.0
DT + OCSVM [123]	16.2	25.9	31.4	34.8	19.0	37.8	39.7	8.4	30.4	26.2	18.1	25.4	26.1
DT + GMM [123]	8.6	28.2	34.8	**23.8**	19.2	**30.8**	27.6	6.3	27.8	22.6	11.4	21.0	21.8
Ours [126]	**3.5**	15.0	**26.9**	31.9	20.8	31.1	**26.7**	**2.9**	**27.2**	21.7	**3.0**	**10.6**	**18.4**

TABLE 5.2: Performance Comparison with the One-Class Domain Adaptation Methods on the CIMN-OCDA Protocol (challenging experimental setting)

Method	HTER (%) ↓												
	C-I	C-M	C-N	I-C	I-M	I-N	M-C	M-I	M-N	N-C	N-I	N-M	Average
DT [60]	54.8	20.6	28.8	58.2	40.0	49.0	42.0	46.3	41.1	39.3	55.2	25.3	41.7
DT + Fine-tune	44.0	22.1	35.2	41.6	23.7	39.3	29.7	36.3	33.0	31.0	29.5	35.2	33.4
DT + OCSVM [123]	23.0	26.0	33.3	**33.4**	27.9	40.8	41.8	9.2	30.6	25.7	23.2	28.8	28.6
DT + GMM [123]	11.7	33.6	34.7	34.7	**20.3**	32.1	32.7	6.9	27.5	22.2	13.5	23.8	24.5
Ours [126]	**3.8**	**17.6**	**28.5**	33.7	28.1	**31.0**	**29.0**	**4.3**	**27.0**	**21.4**	**3.0**	**20.4**	**20.7**

TABLE 5.3: Performance Comparison with the Unsupervised Domain Adaptation Methods

Method	HTER (%) ↓												
	C-I	C-M	C-N	I-C	I-M	I-N	M-C	M-I	M-N	N-C	N-I	N-M	Average
ADDA [129]	41.8	36.6	31.4	49.8	35.1	50.0	39.0	35.2	38.7	28.7	34.6	33.4	37.9
DRCN [130]	44.4	27.6	32.5	48.9	42.0	50.0	28.9	36.8	39.4	32.3	37.4	37.2	38.1
DupGAN [131]	42.4	33.4	30.8	46.5	36.2	47.0	27.1	35.4	34.5	24.6	35.9	33.4	35.6
USDAN-Un [91]	16.0	9.2	/	30.2	25.8	/	13.3	3.4	/	/	/	/	/
KSA [51]	39.3	15.1	31.6	**12.3**	34.9	40.1	**9.1**	33.3	30.4	30.1	38.8	26.1	28.4
ADA [89]	17.5	9.3	29.4	41.5	30.5	41.7	17.7	5.1	32.7	34.1	30.3	31.5	26.8
ML-Net [92]	43.3	14.0	32.4	45.4	35.3	42.8	37.8	11.5	34.6	25.7	30.7	32.6	32.2
UDA [92]	15.6	**9.0**	28.0	34.2	29.0	39.8	16.8	3.0	29.7	**17.9**	23.7	24.4	22.6
Ours [126]	**3.5**	15.0	**26.9**	31.9	**20.8**	**31.1**	26.7	**2.9**	**27.2**	21.7	**3.0**	**10.6**	**18.4**

target dataset as the validation set. The experimental results of the challenging setting are shown in Table 5.2. Compared to the DT method, our proposed method generally reduces the HTER on different tasks by a clear margin. The reduction of average HTER is more than 10%, which validates that our method effectively improves the performance of the face PAD model by using only some genuine face samples in the target domain. Moreover, our method outperforms baseline methods with one-class domain adaptation and achieves the best overall performance under both the ideal and challenging experimental settings [3].

In addition, we also compare our method with state-of-the-art face PAD methods with unsupervised domain adaptation. We find that our proposed method achieves better performance with less target domain data for model training, as shown in

[3]Note that the results of ideal experimental settings are not the upper bound of the challenging experimental setting results. It is because the training data of the SS network in ideal and challenging experimental settings are different.

TABLE 5.4: Performance Comparison with Baseline Methods on the OULU-OCA-SA Protocol

Method	ACER (%) ↓	AUC (%) ↑
DTN [105]	15.61±1.69	/
OCA-FAS [95]	2.26±0.39	/
DT [60]	3.95±0.30	98.17±0.17
Ours [126]	**0.46±0.12**	**99.63±0.12**

TABLE 5.5: Performance (HTER) Comparison with the One-class Domain Adaptation Methods on the WMCA-CI-OCDA Protocol

Methods	HTER (%) ↓						
	Illum. 1	Illum. 2	Illum. 3	Illum. 5	Illum. 6	Illum. 7	Overall
DT [60]	14.17	17.55	42.56	13.24	52.56	15.63	25.95 ± 17.10
DT + Fine-tune	38.95	41.13	46.54	5.97	18.86	21.87	28.89 ± 15.73
DT + OCSVM [92]	12.03	17.09	18.34	3.22	38.90	24.36	18.99 ± 12.05
DT + GMM [92]	16.42	12.31	12.31	14.41	30.41	23.12	18.16 ± 7.21
Ours [126]	13.20	13.55	23.09	0.78	12.22	7.76	**11.77 ± 7.35**

TABLE 5.6: Performance (AUC) Comparison with the One-class Domain Adaptation Methods on the WMCA-CI-OCDA Protocol

Methods	AUC (%) ↑						
	Illum. 1	Illum. 2	Illum. 3	Illum. 5	Illum. 6	Illum. 7	Overall
DT [60]	92.42	92.66	87.99	92.42	76.98	95.10	89.60 ± 6.59
DT + Fine-tune	90.33	84.40	80.24	95.80	94.28	85.96	88.50 ± 6.03
DT + OCSVM [92]	92.82	92.16	95.13	100.00	83.09	97.80	93.50 ± 5.90
DT + GMM [92]	89.74	93.94	94.52	99.23	81.88	98.49	92.97 ± 6.43
Ours [126]	95.10	93.15	90.33	100.00	95.21	97.33	**95.19 ± 3.33**

Table 5.3. Our proposed method achieves the best performance on 8 out of 12 experiments and outperforms the state-of-the-art methods by 4.2% in terms of average HTER. The advantage of our proposed method is especially significant in C-I, I-M, I-N, N-I, and N-M experiments, where our method reduces the HTER by more than 12.1%, 5.0%, 8.7%, 20.7%, and 13.8% compared to the state-of-the-art.

2) Results on the OULU-OCA-SA Protocol:

To compare with OCA-FAS [95], which is a recent method for the face PAD with one-class domain adaptation, we evaluate our proposed method on the OULU-OCA-SA protocol. We conduct experiments on Protocol 3, the most challenging task on OULU-OCA-SA, and the SS model density of our method is set as 10%. From the experimental results shown in Table 5.4, we find that our proposed method outperforms the DTN and OCA-FAS by a clear margin. The ACER and the AUC of our proposed method are 0.46% and 99.63%, respectively. Compared to our baseline method, using the target domain genuine face images for domain adaptation helps to reduce the ACER by 3.49% and improve the AUC by 1.46%.

TABLE 5.7: Performance (HTER) Comparison with the One-class Domain Adaptation Methods on the CIM-N-CS-OCDA Protocol

Protocol	Methods	HTER (%) ↓										
		Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client 7	Client 8	Client 9	Client 10	Overall
I-N-CS	DT [60]	36.16	47.42	30.72	32.71	25.49	47.00	47.95	43.18	41.60	46.51	39.87±8.07
	DT + Fine-tune	22.99	22.52	22.57	16.97	23.35	13.15	35.25	22.42	32.42	24.50	23.61±6.43
	DT + OCSVM [92]	26.55	21.83	15.53	9.19	16.20	20.07	24.41	13.80	16.11	29.49	19.32±6.29
	DT + GMM [92]	24.13	20.38	18.72	10.96	17.56	15.87	26.45	20.71	16.12	25.37	19.63±4.81
	Ours [126]	18.97	18.20	13.05	7.89	16.21	11.53	20.13	14.13	10.22	16.24	**14.66±4.00**
M-N-CS	DT [60]	43.13	38.57	38.49	42.14	37.05	39.49	29.79	30.93	40.25	32.93	37.28±4.60
	DT + Fine-tune	22.32	19.09	14.54	11.04	25.42	25.55	14.49	12.55	12.97	19.64	17.76±5±39
	DT + OCSVM [123]	21.61	18.47	11.68	13.79	13.38	24.56	17.17	13.66	11.41	19.81	16.55±4.48
	DT + GMM [123]	21.50	17.63	11.74	11.64	15.57	19.61	21.36	14.99	11.53	19.92	16.55±4.02
	Ours [126]	12.34	18.06	7.92	8.13	13.41	14.23	13.84	11.01	10.98	16.54	**12.65±3.29**
C-N-CS	DT [60]	31.39	28.11	21.68	32.88	27.54	26.48	29.60	30.35	26.14	28.53	28.27±3.14
	DT + Fine-tune	23.83	23.48	16.19	11.27	24.59	18.73	10.86	15.37	19.39	17.50	18.12±4.90
	DT + OCSVM [123]	23.17	23.03	11.39	7.40	17.67	21.87	13.25	17.80	12.48	25.19	17.33±5.99
	DT + GMM [123]	21.04	20.12	7.16	6.51	20.24	16.20	22.74	19.00	13.82	21.71	16.85±5.90
	Ours [126]	6.22	9.89	3.75	6.07	8.05	10.19	14.94	6.43	10.96	9.14	**8.56±3.18**

TABLE 5.8: Performance (AUC) Comparison with the One-class Domain Adaptation Methods on the CIM-N-CS-OCDA Protocol

Protocol	Methods	AUC (%) ↑										
		Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client 7	Client 8	Client 9	Client 10	Overall
I-N-CS	DT [60]	64.56	44.10	70.95	72.66	79.55	43.58	47.95	57.65	58.00	53.14	59.21±12.48
	DT + Fine-tune	79.98	81.02	82.25	90.19	79.43	93.66	66.37	82.05	69.36	81.16	80.55±8.15
	DT + OCSVM [123]	82.05	85.51	93.19	95.56	90.15	85.59	79.18	91.26	89.59	77.24	86.93±6.07
	DT + GMM [123]	81.30	85.98	89.72	94.29	87.81	88.30	79.97	86.70	88.22	81.94	86.42±4.34
	Ours [126]	89.54	88.92	92.61	96.92	89.05	92.62	86.53	93.29	95.61	92.16	**91.73±3.22**
M-N-CS	DT [60]	58.01	63.94	62.60	60.13	62.98	60.40	77.97	73.60	63.29	68.07	65.10±6.32
	DT + Fine-tune	81.99	84.08	90.84	93.37	79.01	74.73	89.66	93.83	92.48	84.16	86.42±6.60
	DT + OCSVM [123]	85.80	88.63	94.51	93.09	93.93	82.72	90.67	91.02	94.66	85.18	90.02±4.27
	DT + GMM [123]	85.90	89.66	94.91	95.42	90.71	88.22	83.70	92.56	93.29	88.65	90.30±3.83
	Ours [126]	93.83	90.64	97.33	97.70	89.98	93.29	90.32	95.62	95.79	91.74	**93.62±2.90**
C-N-CS	DT [60]	74.22	77.39	86.52	73.49	77.40	78.65	80.52	76.77	78.66	80.63	78.43±3.68
	DT + Fine-tune	82.08	85.69	88.93	91.22	84.28	85.56	89.87	90.25	82.76	88.50	86.91±3.26
	DT + OCSVM [123]	82.89	83.81	95.73	96.16	90.13	86.38	91.41	87.03	91.53	80.66	88.57±5.30
	DT + GMM [123]	85.65	84.87	96.57	96.84	88.41	91.97	80.72	86.35	91.22	86.37	88.90±5.20
	Ours [126]	97.52	95.31	99.18	98.44	96.20	96.12	91.66	98.42	96.72	96.70	**96.63±2.13**

3) Results on the WMCA-CI-OCDA Protocol:

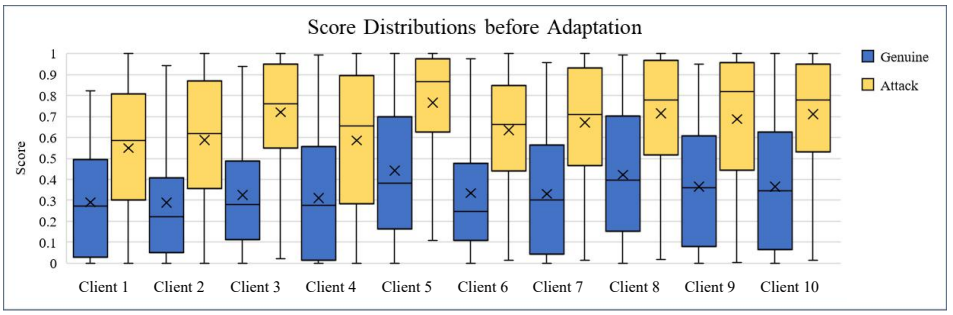
From the experimental results shown in Table 5.5 and Table 5.6, we find that our proposed method also outperforms baseline methods. The advantages of our proposed method are more than 6.3% and 2.6% in terms of the overall HTER and AUC metrics.

5.4.2 Client-Specific One-Class Domain Adaptation Experiments

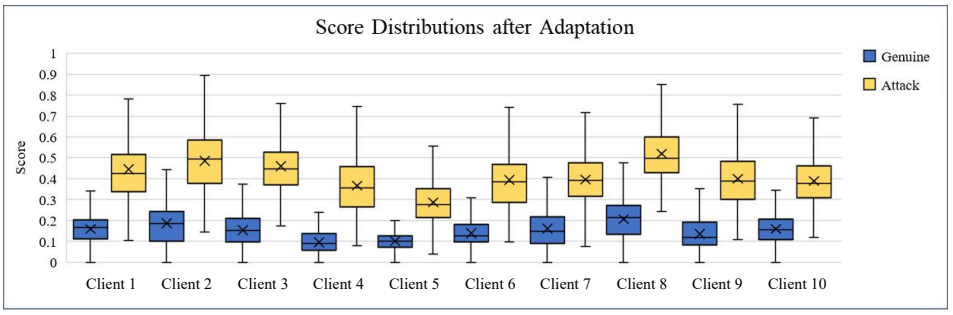
In addition to the experiments under the general one-class domain adaptation setting, we also conduct experiments under the client-specific domain adaptation setting to verify that our proposed method can improve the performance of face PAD for the specific target client by only using a few genuine face images for one-class domain adaptation.

The experiments are conducted on the CIM-N-CS-OCDA protocol. The SS model density is set as 10%, and we evaluate the performance with both HTER and AUC. From the experimental results shown in Table 5.7 and Table 5.8, our method significantly improves the face PAD performance and consistently outperforms baseline methods on all the three sub-protocols. Compared to the DT baseline without using any target domain data for adaptation, our method reduces the average HTER by 25.21%, 24.63%, 19.71% and improves the average AUC by 32.52%, 28.52%, 18.20% on the I-N-CS, M-N-CS, C-N-CS sub-protocols, respectively. Compared to baseline methods that use target domain genuine face samples for model training, our method also shows distinct advantages. On the I-N-CS sub-protocol, our method outperforms baseline methods by more than 4.66% and 4.80% in HTER and AUC. On the M-N-CS and C-N-CS sub-protocols, the improvement are more than 3.90%, 8.29% in terms of HTER, and 3.32%, 7.73% in terms of AUC. Comparing the performance of our proposed method on different sub-protocols, we find that even though we use the same target domain data to train the SS networks, there are distinct performance differences between models with different DT networks. The results indicate that the discrimination ability of the DT's feature representations will influence the final performance of the proposed method.

To intuitively demonstrate the effectiveness of our method, we visualize the score distributions on the C-N-CS sub-protocol. Figure 5.5 (A) shows the score distributions of the DT baseline. The blue and yellow boxes represent the genuine and attack samples, respectively. The distributions of both classes are wide-spreading, and there are obvious overlaps between the genuine face and attack distributions. Figure 5.5 (B) shows the score distributions of our method. With the help of the genuine face samples in the target domain, the score distributions become more compact and separable between genuine and attack samples. As shown in Figure 5.6, we compare our method with other baseline methods using target domain genuine samples for model training. Although the score distributions of the DT + GMM and DT + OCSVM are as compact as ours, our method has the largest separation gap between the distributions of the genuine face and attack samples. Besides, we also visualize the Receiver Operating Characteristic Curves of proposed and baseline methods on the C-N-CS task 1 as in Figure 5.7. As shown in the figure, the True Detection Rate (TDR) of our method is higher than the TDRs of baseline methods at different False Detection Rate (FDR) conditions. The advantage of our method is greater at the thresholds where the FDR is small.



(A) Score Distribution before One-Class Domain Adaptation



(B) Score Distribution after One-Class Domain Adaptation

FIGURE 5.5: The figure shows the score distributions of the DT on 10 tasks of C-N-CS protocol. A pair of blue and yellow boxes are used to illustrate the score distribution for each task. The blue and yellow boxes illustrate the distributions of the testing genuine face and attack samples, respectively.

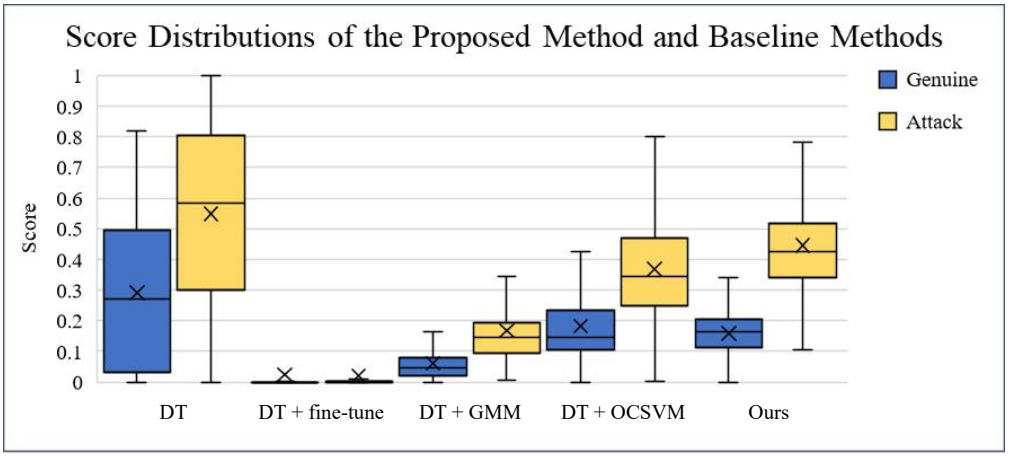


FIGURE 5.6: The figure shows the score distributions of different methods on C-N-CS task 1. From the left to the right are score distributions of the DT, DT+finetune, DT+GMM, DT+OCSVM and our proposed method. The blue and yellow boxes illustrate the distributions of the genuine face and attack samples, respectively. The figure shows that our method has larger separation gap compared to baseline methods.

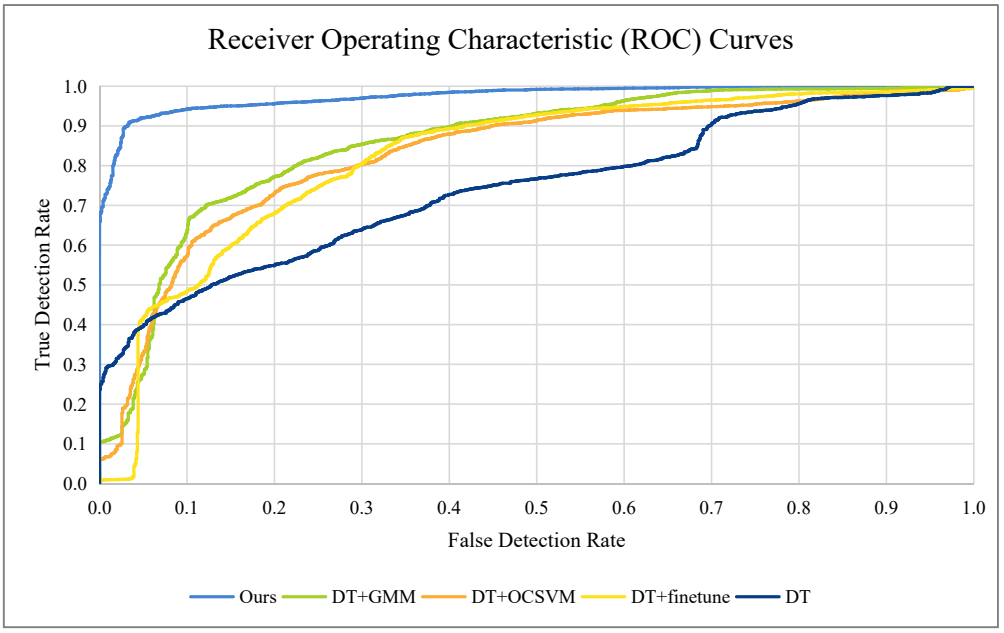


FIGURE 5.7: The figure shows the Receiver Operating Characteristic (ROC) Curves of different methods on C-N-CS task 1. The horizontal axis is the False Detection Rate (FDR) and the vertical axis is the True Detection Rate (TDR).

5.4.3 Ablation Study

Besides, we also conduct experiments to analyze the impacts of the density of SS network, the parameter regrowth mechanism, and different level features.

1) Impact of the Stubborn Student Density:
To study the impact of the SS Density, we conduct experiments on the CIM-N-CS-OCDA protocols. The HTER and AUC at the SS density level 100%, 10%, and 1% are evaluated, and the results are shown in Table 5.9. On the I-N-CS and C-N-CS sub-protocols, the models with the 10% SS density achieve comparable performance to the models with the 100% SS density and perform better than the models with the 1% SS density. On the M-N-CS sub-protocol, the models with 10% and 1% SS density slightly outperform the models with 100% SS density. The results indicate that the SS of 100% density is over-parametric to learn genuine face representations of the DT network. The moderate sparsity of the SS network helps to reduce the model size without severely losing accuracy. To verify the feasibility of model compression, we measure the model size of the SS at different density levels. As shown in Table 5.10, the number of non-zero parameters at 100%, 10%, and 1% density are 1.73 M, 0.18 M, and 0.02 M. We save the sparse models in Coordinate (COO) data format and the size of the SS model with 100%,

TABLE 5.9: Performance of the Proposed Method with Different Stubborn Student Density on the CIM-N-CS-OCDA Protocol

Protocol	SS Density (%)	HTER (%) ↓	AUC (%) ↑
I-N-CS	100	15.19	91.26
	10	**14.66**	**91.73**
	1	16.94	89.26
M-N-CS	100	14.62	92.60
	10	**12.65**	**93.62**
	1	13.16	92.85
C-N-CS	100	8.65	96.50
	10	**8.56**	**96.63**
	1	10.71	95.03

TABLE 5.10: Model Size of the Stubborn Student at Different Density

SS Density (%)	No. Non-zero Params (M)	Memory (MB)
100	1.73	6.62
10	0.18	2.02
1	0.02	0.25

TABLE 5.11: Performance of the Proposed Method with Single-Level and Multi-Level Distillation on the CIM-N-CS-OCDA Protocol

Protocol	Distillation Level	HTER (%) ↓	AUC (%) ↑
I-N-CS	Single-Level	**14.15**	**92.37**
	Multi-Level	14.66	91.73
M-N-CS	Single-Level	14.68	92.66
	Multi-Level	**12.65**	**93.62**
C-N-CS	Single-Level	10.34	95.41
	Multi-Level	**8.56**	**96.63**

10%, and 1% density are 6.62 MB, 2.02 MB, and 0.25 MB, respectively. Note that the compression rate of the model size in terms of memory is not the same as the number of non-zero parameters. It is because that we need to store the coordinates for the non-zero parameters of the sparse model.

2) Impact of the Parameter Regrowth Mechanism:
To analyze the impact of the parameter regrowth mechanism, we conduct ablation experiments on the C-N-CS sub-protocol and plot the changing of the average HTER with the number of training iterations in Figure 5.8. We can see that the parameter regrowth mechanism helps optimize the training process, especially when the SS network is at a lower density.

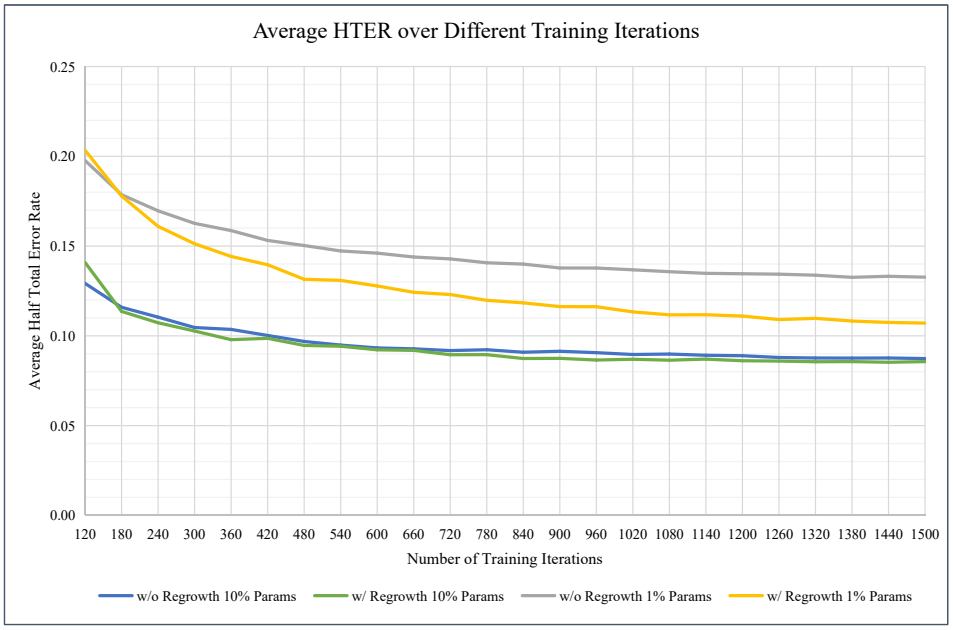
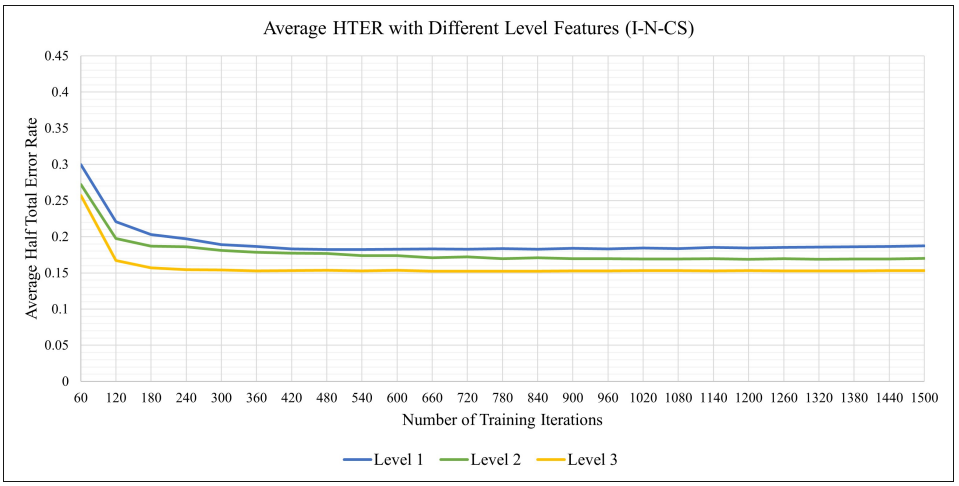


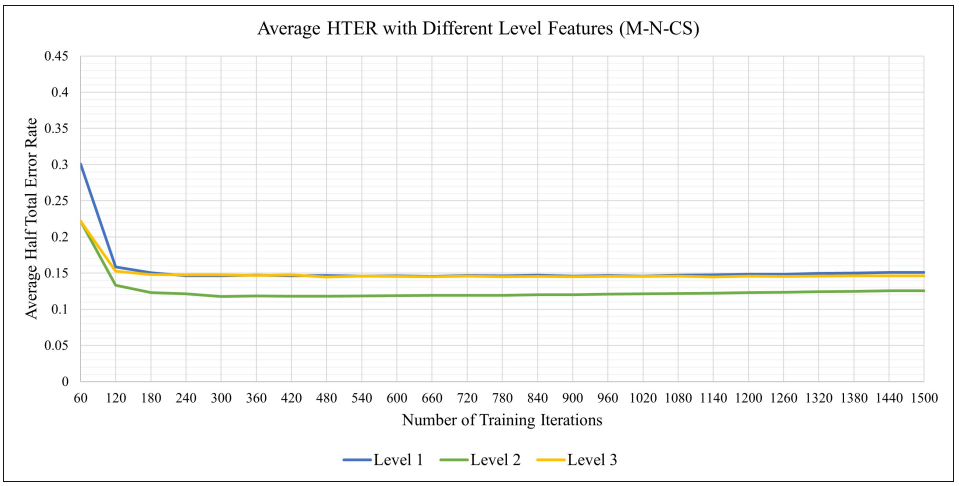
FIGURE 5.8: This figure shows the average HTER of our proposed method over the number of training iterations on C-N-CS tasks. The four curves illustrate the impact of the regrowth mechanism on the proposed method at two SS model density conditions.

3) Impact of the Feature Level:

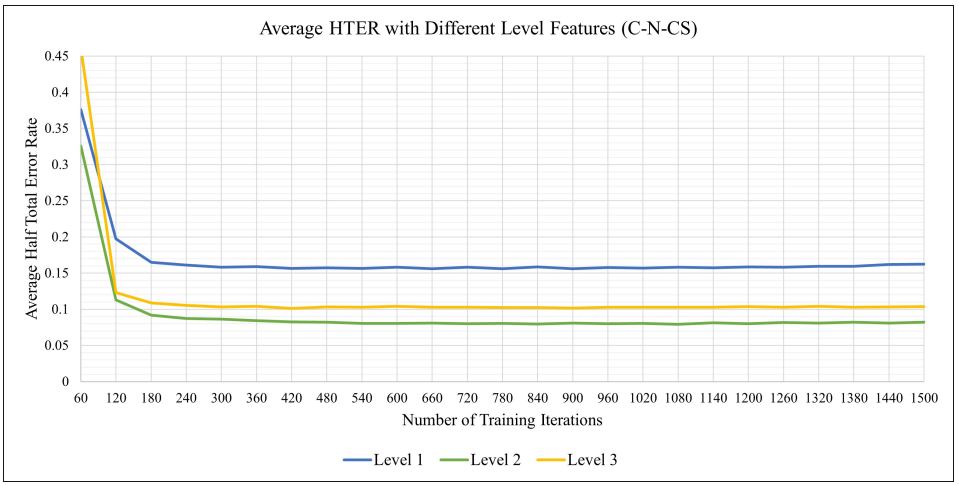
To analyze the impact of different level features, we conduct comparison experiments on the CIM-N-CS-OCDA protocols, and the results are shown in Table 5.11. On the I-N-CS sub-protocol, our method with single-level and multi-level distillation achieve comparable results in the HTER, but the single-level distillation one has better AUC performance. On the M-N-CS and C-N-CS sub-protocols, our method with multi-level distillation outperforms the single-level one by 2.03%, 1.78% in HTER and 0.96%, 1.22% in AUC. The experimental results show that using the aggregation of three level features may not outperform the single-level features. We visualize the average HTER performance with different level features as shown in Figure 5.9. Level 1 feature performs the worst on all protocols. Level 3 feature performs the best on I-N-CS protocol while Level 2 feature performs the best on M-N-CS and C-N-CS protocols. The visualization results show that different level features have different discrimination abilities and some of them outperform the others. Therefore, it is promising to explore the feature selection in future work to further improve the performance.



(A) I-N-CS



(B) M-N-CS



(C) C-N-CS

FIGURE 5.9: The figure (a), (b), (c) show the average HTER with different level features pretrained on I, M, C dataset, respectively. The average HTER performance with Level 1, Level 2, and Level 3 features are plotted in blue, green, and yellow, respectively.

(A) correctly classified genuine samples

(C) correctly classified attack samples

(B) misclassified genuine samples

(D) misclassified attack samples

FIGURE 5.10: The figures show examples of the image samples correctly classified and misclassified by our model on C-N-CS task 1. Figure (A) shows the correctly classified genuine face samples; figure (B) shows the misclassified genuine face samples; figure (C) shows the correctly classified attack samples; figure (D) shows the misclassified attack samples.

5.4.4 Discussion

Although our proposed method generally outperforms prior methods, there is still room for performance improvement under the general and client-specific one-class domain adaptation setting. The HTER performance is not satisfactory in some experimental settings, especially where a simple dataset is used as the source domain and a more complex dataset as the target domain, such as I-C, I-N, and M-N in Table 5.1.

Besides, the classification accuracy of our model for some difficult samples needs to be further improved. As shown in Figure 5.10, some genuine face samples with unusual facial expressions or backgrounds are misclassified as attacks; some attack samples with natural skin color and higher definition are misclassified as genuine face images.

From the experimental results in Table 5.7, Table 5.8, and Table 5.11, even using the same target domain data to train the student network SS, the performance of the proposed method varies with the feature quality of the teacher network DT. Therefore, a promising direction for future work is to further improve the feature learning and feature selection of the teacher network.

5.5 Chapter Summary

In this paper, we introduce a framework to address the cross-domain problem in face PAD with one-class domain adaptation, which improves the cross-domain performance of the face PAD model by utilizing only a few genuine face samples collected in the target domain. Under the framework, a teacher network is trained with genuine face and attack samples of the source domain to provide multi-level discriminative feature representations for face PAD. Student networks are trained with only genuine face samples of the target domain to generate similar representations to the teacher's outputs. To verify the effectiveness of our method under one-class domain adaptation settings, we devised two new protocols on public face PAD datasets and did extensive experiments. The experimental results show that our method outperforms baseline methods under one-class domain adaptation settings and even performs better than state-of-the-art methods with unsupervised domain adaptation. However, there is still room for performance improvement in some experimental settings. In future work, we will explore methods to further improve the feature learning and feature selection of the teacher network.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we study the cross-domain problems in face PAD and propose several methods which apply to different application scenarios. In consideration of the difference between genuine face and attack samples, such as the feasibility and the expense of data collection in practical scenarios, our methods are devised with more attention to genuine face samples.

Considering that the attackers may launch novel presentation attacks which are unseen to the face PAD model at the training phase, in our first work presented in Chapter 3, we explore the unseen attack problem in face PAD. Different from prior works, we propose a face PAD method based on deep metric learning, which detects attacks directly on the learned feature space with no need for additional classifiers to be trained. The mapping from the image space to the target feature space is implemented with a CNN-based feature extractor. To force the genuine face samples to maintain intra-class compactness and ensure inter-class separation from the attack samples, we devise a hypersphere loss function to direct the optimization of the feature extractor. To verify the effectiveness of the proposed method, we did extensive experiments on multiple prevailing datasets. The experimental results demonstrate that our method is effective in detecting unseen attacks and outperforms prior methods. However, the practical application scenarios of face PAD are more complex and challenging than the laboratory unseen attack setting. Beyond the threats of unseen attacks encountered during the testing phase, the changes

in illumination conditions and camera sensors will also degrade the accuracy and reliability of the face PAD systems.

Recently, multi-modality face PAD methods have shown promising results with additional sensors for information acquisition, while their generalization problems are not well explored. In our second work presented in Chapter 4, we study the generalization problems of face PAD in bi-modality scenarios and propose a method that generalizes better to unseen attack and illumination variations. Different from prior works, our method explicitly connects face images of different modalities via asymmetric modality translation. In this method, we devise an asymmetric modality translator which successfully translates genuine face images from the source modality to the target modality while failing for attacks. The discrepancy of modality translation between genuine faces and attack samples is utilized as an effective clue for discriminating various spoofing faces from genuine faces. Besides, an illumination normalization (IN) module based on PLGF descriptor is used to alleviate the interference of the illumination variations on sensitive modalities. Extensive experimental results show that our method applies to different modality settings and can effectively detect various seen and unseen attacks under varying illumination conditions. However, the performance of our method in cross-dataset evaluation is not satisfactory.

Domain adaptation is a typical approach to improving the cross-domain performance of face PAD with the help of target domain data. However, it has always been a non-trivial challenge to collect sufficient data samples in the target domain, especially for attack samples. Compared to attack samples, genuine face samples are much easier and cheaper to collect. Therefore, in the third work presented in Chapter 5, we study the one-class domain adaptation problem in face PAD which aims to improve the cross-domain performance of the face PAD with the help of a few genuine face samples collected in the target domain. We propose a method by introducing teacher-student learning to address the one-class domain adaptation problem in face PAD. A teacher network is trained with source domain samples to provide discriminative feature representations for face PAD. Student networks are trained to mimic the teacher network and learn similar representations for genuine face samples of the target domain. In the test phase, the similarity score between the representations of the teacher and student networks is used to distinguish attacks from genuine ones. To verify the effectiveness of the proposed

method, we devise two new protocols for the evaluation of the face PAD model under the one-class domain adaptation scenarios and conduct extensive experiments on multiple datasets. The experimental results show that our method outperforms baseline methods with one-class domain adaptation and also unsupervised domain adaptation techniques.

6.2 Future Work

Based on the experimental results in Chapter 3, the performance of our method under the unseen attack setting is not always satisfactory, especially at the scenarios where high-fidelity mask, makeup, and some partial attacks are treated as unseen attacks for evaluation. The poor performance is caused by insufficient attack samples for model training. Therefore, one of the possible direction for future work is to address the insufficiency of training data with the help of advanced generative models.

From the experimental results in Chapter 4, although multi-modality face PAD methods show advantages in improving the generalization performance against unseen attacks and illumination variation, the cross-dataset generalization problem on multi-modality face PAD may not be easier than single VIS-modality ones. It is because the specifications of multi-modality sensors and the data preprocessing schemes are different in different datasets. In practical application, the data preprocessing scheme could be easily unified to protect the performance of face PAD from being severely degraded. However, the cross-sensor generalization problem is crucial and worthy of attention since more sensor devices are used in multi-modality face PAD than single-modality ones. Besides, the generalization performance of VIS-modality face PAD has been improved with the help of data augmentation techniques, while the data augmentation techniques for multi-modality face PAD have been scarcely studied.

Domain generalization and adaptation problems have been extensively studied in recent VIS-modality face PAD research. Although our proposed method in Chapter 5 generally outperforms prior methods, there is still room for performance improvement when the target domain is more complex than the source domain. Moreover, even using the same target domain data to train the student network,

the performance of the proposed method varies with the feature quality of the teacher network. Therefore, there is a need to improve the feature learning and feature selection of the teacher network. Besides, most recent face PAD models are jointly trained with data samples from multiple data domains to improve the generalization ability. However, the data samples from different data sources are diverse in quantity and quality. The domain imbalance problem and the robustness of jointly training to noisy or even poison data are also worthy of exploration in future work.

List of Author's Awards, Patents, and Publications

Journal Articles

- **Zhi Li**, Rizhao Cai, Haoliang Li, Kwok-Yan Lam, Yongjian Hu, and Alex C. Kot, "One-Class Knowledge Distillation for Face Presentation Attack Detection," in *IEEE Transactions on Information Forensics and Security, doi: 10.1109/TIFS.2022.3178240.*

- **Zhi Li**, Haoliang Li, Xin Luo, Yongjian Hu, Kwok-Yan Lam, and Alex C. Kot, "Asymmetric Modality Translation for Face Presentation Attack Detection," in *IEEE Transactions on Multimedia, doi: 10.1109/TMM.2021.3121140.*

Conference Proceedings

- **Zhi Li**, Haoliang Li, Kwok-Yan Lam, and Alex C. Kot, "Unseen Face Presentation Attack Detection with Hypersphere Loss," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.*

Bibliography

[1] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, 2021. 1

[2] R. Ramachandra and C. Busch, "Presentation attack detection methods for face recognition systems: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 1, pp. 1–37, 2017. 2, 16

[3] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, "Oulu-npu: A mobile face presentation attack database with real-world variations," in *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*. IEEE, 2017, pp. 612–618. 2, 9, 14, 19, 26, 27, 28, 29, 40, 59, 69

[4] Z. Boulkenafet, J. Komulainen, Z. Akhtar, A. Benlamoudi, D. Samai, S. E. Bekhouche, A. Ouafi, F. Dornaika, A. Taleb-Ahmed, L. Qin *et al.*, "A competition on generalized software-based face presentation attack detection in mobile scenarios," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2017, pp. 688–696. 2, 59

[5] I. Chingovska, N. Erdogmus, A. Anjos, and S. Marcel, "Face recognition systems under spoofing attacks," in *Face Recognition Across the Imaging Spectrum*. Springer, 2016, pp. 165–194. 5, 15, 42

[6] S. Zhang, X. Wang, A. Liu, C. Zhao, J. Wan, S. Escalera, H. Shi, Z. Wang, and S. Z. Li, "A dataset and benchmark for large-scale multi-modal face anti-spoofing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 919–928. 5, 12, 15, 33, 35, 42, 43, 44, 45, 50, 51

[7] A. Liu, Z. Tan, J. Wan, S. Escalera, G. Guo, and S. Z. Li, "Casia-surf cefa: A benchmark for multi-modal cross-ethnicity face anti-spoofing," in *Proceedings*

of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1179–1187. 12

[8] A. George, Z. Mostaani, D. Geissenbuhler, O. Nikisins, A. Anjos, and S. Marcel, "Biometric face presentation attack detection with multi-channel convolutional neural network," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 42–55, 2019. 5, 12, 15, 33, 35, 42, 44, 45, 47, 48, 50

[9] G. Heusch, A. George, D. Geissbühler, Z. Mostaani, and S. Marcel, "Deep models and shortwave infrared information to detect face presentation attacks," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 4, pp. 399–409, 2020. 5, 12, 44, 45, 47, 48, 49, 50, 51, 55

[10] D. Siegmund, F. Kerckhoff, J. Y. Magdaleno, N. Jansen, F. Kirchbuchner, and A. Kuijper, "Face presentation attack detection in ultraviolet spectrum via local and global features," in *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 2020, pp. 1–5. 5

[11] R. Raghavendra, K. B. Raja, and C. Busch, "Presentation attack detection for face recognition using light field camera," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1060–1075, 2015. 5

[12] A. Sepas-Moghaddam, F. Pereira, and P. L. Correia, "Light field-based face presentation attack detection: reviewing, benchmarking and one step further," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1696–1709, 2018. 5

[13] X. Wu, J. Zhou, J. Liu, F. Ni, and H. Fan, "Single-shot face anti-spoofing for dual pixel camera," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1440–1451, 2020. 5

[14] J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of fourier spectra," in *Biometric technology for human identification*, vol. 5404. SPIE, 2004, pp. 296–303. 6

[15] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *European Conference on Computer Vision*. Springer, 2010, pp. 504–517. 7

[16] B. Peixoto, C. Michelassi, and A. Rocha, "Face liveness detection under bad illumination conditions," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 3557–3560. 6

[17] J. Galbally, S. Marcel, and J. Fierrez, "Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition," *IEEE transactions on image processing*, vol. 23, no. 2, pp. 710–724, 2013. 6, 25

[18] J. Galbally and S. Marcel, "Face anti-spoofing based on general image quality assessment," in *2014 22nd international conference on pattern recognition*. IEEE, 2014, pp. 1173–1178. 7, 19, 45

[19] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746–761, 2015. 6, 7, 14, 19, 25, 45, 69

[20] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *2011 international joint conference on Biometrics (IJCB)*. IEEE, 2011, pp. 1–7. 6

[21] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *2012 BIOSIG-proceedings of the international conference of biometrics special interest group (BIOSIG)*. IEEE, 2012, pp. 1–7. 13, 25, 69

[22] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analysis," in *2015 IEEE international conference on image processing (ICIP)*. IEEE, 2015, pp. 2636–2640. 45

[23] A. Agarwal, R. Singh, and M. Vatsa, "Face anti-spoofing using haralick features," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2016, pp. 1–6. 6, 7, 45

[24] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *2007 IEEE 11th international conference on computer vision*. IEEE, 2007, pp. 1–8. 6, 7

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

IEEE Transactions on Information Forensics and Security, vol. 2, no. 3, pp. 548–558, 2007. 6, 7

[26] K. Kollreider, H. Fronthaler, and J. Bigun, "Evaluating liveness by face images and the structure tensor," in *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*. IEEE, 2005, pp. 75–80. 7

[27] ——, "Non-intrusive liveness detection by face images," *Image and Vision Computing*, vol. 27, no. 3, pp. 233–244, 2009. 6

[28] T. d. Freitas Pereira, J. Komulainen, A. Anjos, J. M. De Martino, A. Hadid, M. Pietikäinen, and S. Marcel, "Face liveness detection using dynamic texture," *EURASIP Journal on Image and video Processing*, vol. 2014, no. 1, pp. 1–15, 2014. 6, 7

[29] S. R. Arashloo and J. Kittler, "Dynamic texture recognition using multiscale binarized statistical image features," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2099–2109, 2014.

[30] S. R. Arashloo, J. Kittler, and W. Christmas, "Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2396–2407, 2015. 26

[31] A. Pinto, H. Pedrini, W. R. Schwartz, and A. Rocha, "Face spoofing detection through visual codebooks of spectral temporal cubes," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4726–4740, 2015. 7

[32] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. Ho, "Detection of face spoofing using visual dynamics," *IEEE transactions on information forensics and security*, vol. 10, no. 4, pp. 762–777, 2015. 6

[33] S. Liu, P. C. Yuen, S. Zhang, and G. Zhao, "3d mask face anti-spoofing with remote photoplethysmography," in *European Conference on Computer Vision*. Springer, 2016, pp. 85–100. 6

[34] X. Li, J. Komulainen, G. Zhao, P.-C. Yuen, and M. Pietikäinen, "Generalized face anti-spoofing by detecting pulse from face videos," in *2016 23rd*

International Conference on Pattern Recognition (ICPR). IEEE, 2016, pp. 4244–4249.

[35] S.-Q. Liu, X. Lan, and P. C. Yuen, "Remote photoplethysmography correspondence feature for 3d mask face presentation attack detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 558–573.

[36] ——, "Multi-channel remote photoplethysmography correspondence feature for 3d mask face presentation attack detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2683–2696, 2021. 6, 7

[37] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995. 7

[38] J. Liu, J. Chen, and J. Ye, "Large-scale sparse logistic regression," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 547–556. 7

[39] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2. 7

[40] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997. 7

[41] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Advances in neural information processing systems*, vol. 12, 1999. 7, 25, 72

[42] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977. 7, 72

[43] S. R. Arashloo, J. Kittler, and W. Christmas, "An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol," *IEEE access*, vol. 5, pp. 13 868–13 882, 2017. 7, 19, 25, 26, 27, 28, 31

[44] O. Nikisins, A. Mohammadi, A. Anjos, and S. Marcel, "On effectiveness of anomaly detection approaches against unseen presentation attacks in face anti-spoofing," in *2018 International Conference on Biometrics (ICB)*. IEEE, 2018, pp. 75–81. 19

[45] F. Xiong and W. AbdAlmageed, "Unknown presentation attack detection with face rgb images," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2018, pp. 1–9. 7, 19, 25, 26, 27, 28, 31

[46] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *arXiv preprint arXiv:1408.5601*, 2014. 7

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 7

[48] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5295–5305. 8, 12, 34, 45

[49] Z. Yu, J. Wan, Y. Qin, X. Li, S. Z. Li, and G. Zhao, "Nas-fas: Static-dynamic central difference network search for face anti-spoofing," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 9, pp. 3005–3023, 2020. 8

[50] Z. Yu, Y. Qin, X. Xu, C. Zhao, Z. Wang, Z. Lei, and G. Zhao, "Auto-fas: Searching lightweight networks for face anti-spoofing," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 996–1000. 8

[51] H. Li, W. Li, H. Cao, S. Wang, F. Huang, and A. C. Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1794–1809, 2018. 8, 11, 14, 69, 70, 72, 74

[52] R. Shao, X. Lan, and P. C. Yuen, "Deep convolutional dynamic texture learning with adaptive channel-discriminability for 3d mask face anti-spoofing," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2017, pp. 748–755. 8

[53] H. Li, P. He, S. Wang, A. Rocha, X. Jiang, and A. C. Kot, "Learning generalized deep feature representation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2639–2652, 2018. 9

[54] R. Shao, X. Lan, and P. C. Yuen, "Joint discriminative learning of deep dynamic textures for 3d mask face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 923–938, 2018.

[55] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3507–3516.

[56] Z. Wang, Z. Yu, C. Zhao, X. Zhu, Y. Qin, Q. Zhou, F. Zhou, and Z. Lei, "Deep spatial gradient and temporal depth learning for face anti-spoofing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5042–5051.

[57] W. Zheng, M. Yue, S. Zhao, and S. Liu, "Attention-based spatial-temporal multi-scale network for face anti-spoofing," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 3, pp. 296–307, 2021.

[58] Z. Wang, Q. Wang, W. Deng, and G. Guo, "Learning multi-granularity temporal characteristics for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1254–1269, 2022. 8

[59] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2017, pp. 319–328. 8

[60] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 389–398. 8, 26, 28, 29, 64, 71, 74, 75, 76

[61] Y. Zhang, Z. Yin, Y. Li, G. Yin, J. Yan, J. Shao, and Z. Liu, "Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations," in *European Conference on Computer Vision*. Springer, 2020, pp. 70–85. 8

[62] Z. Yu, X. Li, X. Niu, J. Shi, and G. Zhao, "Face anti-spoofing with human material perception," in *European conference on computer vision*. Springer, 2020, pp. 557–575. 8

[63] Z. Yu, X. Li, J. Shi, Z. Xia, and G. Zhao, "Revisiting pixel-wise supervision for face anti-spoofing," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 3, pp. 285–295, 2021. 8

[64] 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 *European Conference on Computer Vision*. Springer, 2020, pp. 641–657. 8

[65] W. Sun, Y. Song, C. Chen, J. Huang, and A. C. Kot, "Face spoofing detection based on local ternary label supervision in fully convolutional networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3181–3196, 2020. 8

[66] A. George and S. Marcel, "Deep pixel-wise binary supervision for face presentation attack detection," in *2019 International Conference on Biometrics (ICB)*. IEEE, 2019, pp. 1–8. 8, 12, 33, 40, 44, 64

[67] A. Jourabloo, Y. Liu, and X. Liu, "Face de-spoofing: Anti-spoofing via noise modeling," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 290–306. 8

[68] Y. Liu, J. Stehouwer, and X. Liu, "On disentangling spoof trace for generic face anti-spoofing," in *European Conference on Computer Vision*. Springer, 2020, pp. 406–422. 8

[69] Y. Liu and X. Liu, "Spoof trace disentanglement for generic face anti-spoofing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 8

[70] A. Pinto, S. Goldenstein, A. Ferreira, T. Carvalho, H. Pedrini, and A. Rocha, "Leveraging shape, reflectance and albedo from shading for face presentation attack detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3347–3358, 2020. 8

[71] J. M. Di Martino, Q. Qiu, and G. Sapiro, "Rethinking shape from shading for spoofing detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 1086–1099, 2020. 8

[72] A. F. Ebihara, K. Sakurai, and H. Imaoka, "Efficient face spoofing detection with flash," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 4, pp. 535–549, 2021. 8

[73] X. Zhu, S. Li, X. Zhang, H. Li, and A. C. Kot, "Detection of spoofing medium contours for face anti-spoofing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 2039–2045, 2019. 9

[74] R. Cai, H. Li, S. Wang, C. Chen, and A. C. Kot, "Drl-fas: A novel framework based on deep reinforcement learning for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 937–951, 2020. 9

[75] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8695–8704. 9

[76] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International conference on machine learning*. PMLR, 2020, pp. 3247–3258. 9

[77] B. Chen, W. Yang, and S. Wang, "Face anti-spoofing by fusing high and low frequency features for advanced generalization capability," in *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2020, pp. 199–204. 9

[78] B. Chen, W. Yang, H. Li, S. Wang, and S. Kwong, "Camera invariant feature learning for generalized face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2477–2492, 2021. 9

[79] M. Fang, N. Damer, F. Kirchbuchner, and A. Kuijper, "Learnable multi-level frequency decomposition and hierarchical attention mechanism for generalized face presentation attack detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 3722–3731. 9

[80] G. Wang, H. Han, S. Shan, and X. Chen, "Cross-domain face presentation attack detection via multi-domain disentangled representation learning," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6678–6687. 10

[81] J. Wang, Z. Zhao, W. Jin, X. Duan, Z. Lei, B. Huai, Y. Wu, and X. He, "Vlad-vsa: Cross-domain face presentation attack detection with vocabulary separation and adaptation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1497–1506. 10

[82] S. Liu, K.-Y. Zhang, T. Yao, K. Sheng, S. Ding, Y. Tai, J. Li, Y. Xie, and L. Ma, "Dual reweighting domain generalization for face presentation attack detection," *arXiv preprint arXiv:2106.16128*, 2021. 10

[83] 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," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 916–11 923. 10

[84] R. Shao, X. Lan, and P. C. Yuen, "Regularized fine-grained meta face anti-spoofing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 974–11 981. 10

[85] S. Liu, K.-Y. Zhang, T. Yao, M. Bi, S. Ding, J. Li, F. Huang, and L. Ma, "Adaptive normalized representation learning for generalizable face anti-spoofing," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1469–1477. 10

[86] Y. Qin, Z. Yu, L. Yan, Z. Wang, C. Zhao, and Z. Lei, "Meta-teacher for face anti-spoofing," *IEEE transactions on pattern analysis and machine intelligence*, 2021. 10, 60

[87] R. Cai, Z. Li, R. Wan, H. Li, Y. Hu, and A. C. Kot, "Learning meta pattern for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1201–1213, 2022. 10

[88] Z. Chen, T. Yao, K. Sheng, S. Ding, Y. Tai, J. Li, F. Huang, and X. Jin, "Generalizable representation learning for mixture domain face anti-spoofing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1132–1139. 10

[89] G. Wang, H. Han, S. Shan, and X. Chen, "Improving cross-database face presentation attack detection via adversarial domain adaptation," in *2019 International Conference on Biometrics (ICB)*. IEEE, 2019, pp. 1–8. 11, 70, 72, 74

[90] H. Li, S. Wang, P. He, and A. Rocha, "Face anti-spoofing with deep neural network distillation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 933–946, 2020. 11, 60, 66

[91] Y. Jia, J. Zhang, S. Shan, and X. Chen, "Unified unsupervised and semi-supervised domain adaptation network for cross-scenario face anti-spoofing," *Pattern Recognition*, vol. 115, p. 107888, 2021. 11, 72, 74

[92] G. Wang, H. Han, S. Shan, and X. Chen, "Unsupervised adversarial domain adaptation for cross-domain face presentation attack detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 56–69, 2020. 11, 70, 72, 74, 75, 76

[93] J. Yang, Z. Lei, D. Yi, and S. Z. Li, "Person-specific face antispoofing with subject domain adaptation," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 797–809, 2015. 11, 60

[94] A. Mohammadi, S. Bhattacharjee, and S. Marcel, "Domain adaptation for generalization of face presentation attack detection in mobile settengs with minimal information," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 1001–1005. 11, 60

[95] Y. Qin, W. Zhang, J. Shi, Z. Wang, and L. Yan, "One-class adaptation face anti-spoofing with loss function search," *neurocomputing*, vol. 417, pp. 384–395, 2020. 11, 60, 70, 72, 75

[96] A. George and S. Marcel, "Learning one class representations for face presentation attack detection using multi-channel convolutional neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 361–375, 2020. 12, 33, 35

[97] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 12, 24, 27, 33, 37

[98] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141. 12, 33

[99] S. Zhang, A. Liu, J. Wan, Y. Liang, G. Guo, S. Escalera, H. J. Escalante, and S. Z. Li, "Casia-surf: A large-scale multi-modal benchmark for face anti-spoofing," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 2, pp. 182–193, 2020. 12, 15, 33, 35, 42, 43, 44, 45, 50, 51

[100] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018. 12

[101] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708. 12, 40, 45

[102] Z. Yu, Y. Qin, X. Li, Z. Wang, C. Zhao, Z. Lei, and G. Zhao, "Multi-modal face anti-spoofing based on central difference networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 650–651. 12, 34, 35, 44, 45, 47, 48, 49, 50, 51, 55

[103] A. Liu, Z. Tan, J. Wan, Y. Liang, Z. Lei, G. Guo, and S. Z. Li, "Face anti-spoofing via adversarial cross-modality translation," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2759–2772, 2021. 12

[104] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in *2012 5th IAPR international conference on Biometrics (ICB)*. IEEE, 2012, pp. 26–31. 13, 25, 69

[105] Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu, "Deep tree learning for zero-shot face anti-spoofing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4680–4689. 14, 20, 24, 25, 26, 27, 28, 29, 31, 75

[106] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982. 16

[107] S. Fatemifar, S. R. Arashloo, M. Awais, and J. Kittler, "Spoofing attack detection by anomaly detection," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8464–8468. 19

[108] Z. Li, H. Li, K.-Y. Lam, and A. C. Kot, "Unseen face presentation attack detection with hypersphere loss," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2852–2856. 20, 28

[109] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International workshop on similarity-based pattern recognition*. Springer, 2015, pp. 84–92. 23

[110] J. Kannala and E. Rahtu, "Bsif: Binarized statistical image features," in *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*. IEEE, 2012, pp. 1363–1366. 26

[111] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002. 26

[112] Z. Li, H. Li, X. Luo, Y. Hu, K.-Y. Lam, and A. C. Kot, "Asymmetric modality translation for face presentation attack detection," *IEEE Transactions on Multimedia*, 2021. 34, 47, 48, 49, 50, 51, 53, 55, 56, 57, 66

[113] J. Huang, J. Liao, and S. Kwong, "Semantic example guided image-to-image translation," *IEEE Transactions on Multimedia*, vol. 23, pp. 1654–1665, 2020. 36

[114] H. Zhan, C. Yi, B. Shi, J. Lin, L.-Y. Duan, and A. C. Kot, "Pose-normalized and appearance-preserved street-to-shop clothing image generation and feature learning," *IEEE Transactions on Multimedia*, vol. 23, pp. 133–144, 2020.

[115] J. Song, J. Zhang, L. Gao, Z. Zhao, and H. T. Shen, "Agegan++: Face aging and rejuvenation with dual conditional gans," *IEEE Transactions on Multimedia*, vol. 24, pp. 791–804, 2021.

[116] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 661–18 673, 2020. 37

[117] X. Zhang, Y. Zhu, W. Chen, W. Liu, and L. Shen, "Gated switchgan for multi-domain facial image translation," *IEEE Transactions on Multimedia*, 2021.

[118] H. Emami, M. M. Aliabadi, M. Dong, and R. B. Chinnam, "Spa-gan: Spatial attention gan for image-to-image translation," *IEEE Transactions on Multimedia*, vol. 23, pp. 391–401, 2020. 36

[119] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134. 37, 39

[120] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016. 37

[121] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018. 37

[122] D. Bhattacharjee and H. Roy, "Pattern of local gravitational force (plgf): A novel local image descriptor," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 2, pp. 595–607, 2019. 40

[123] S. Fatemifar, S. R. Arashloo, M. Awais, and J. Kittler, "Client-specific anomaly detection for face presentation attack detection," *Pattern Recognition*, vol. 112, p. 107696, 2021. 60, 72, 74, 76

[124] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4183–4192. 60, 63

[125] T. Dettmers and L. Zettlemoyer, "Sparse networks from scratch: Faster training without losing performance," *arXiv preprint arXiv:1907.04840*, 2019. 62, 63, 66, 67, 68

[126] Z. Li, R. Cai, H. Li, K.-Y. Lam, Y. Hu, and A. C. Kot, "One-class knowledge distillation for face presentation attack detection," *IEEE Transactions on Information Forensics and Security*, 2022. 62, 74, 75, 76

[127] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019. 68, 73

[128] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 73

[129] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176. 74

[130] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 597–613. 74

[131] L. Hu, M. Kan, S. Shan, and X. Chen, "Duplex generative adversarial network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1498–1507. 74