System reliability enhancement via deep-driven computer vision

Ding, Shuya

2021

Ding, S. (2021). System reliability enhancement via deep-driven computer vision. Doctoral thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/151578>

<https://hdl.handle.net/10356/151578>

<https://doi.org/10.32657/10356/151578>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 08:18:19 SGT

SYSTEM RELIABILITY ENHANCEMENT VIA DEEP-DRIVEN COMPUTER VISION

Student: Shuya Ding

Supervisor: Assoc Prof. Jun Luo

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy (Ph.D)

2021

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.

15 Jan 2021

.....
Date

.....
Shuya Ding

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.

15 Jan 2021
.....
Date

.....
Jun Luo

Authorship Attribution Statement

This thesis contains material from three papers accepted (submitted) at journals/conferences in which I am listed as first author.

Chapter 3 is published as **Guanlin Li*, Shuya Ding*, Jun Luo, and Chang Liu. "Enhancing Intrinsic Adversarial Robustness via Feature Pyramid Decoder," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).**

The contributions of the co-authors are as follows:

- Mr. Guanlin Li and I contributed equally to this paper.
- I co-worked with Mr. Guanlin Li to decide research direction and design framework.
- Mr. Guanlin Li, Mr. Chang Liu, and I performed experiments.
- I provided experimental analysis.
- I wrote manuscript drafts. The manuscript was revised together with Assoc Prof. Jun Luo.

Chapter 4 is published as **Shuya Ding*, Zhe Chen*, Tianyue Zheng, and Jun Luo. "RF-Net: A Unified Meta-Learning Framework for RF-Enabled One-Shot Human Activity Recognition," in 2020 ACM Conference on Embedded Networked Sensor Systems (SenSys).**

The contributions of the co-authors are as follows:

- Dr. Zhe Chen and I contributed equally to this paper.
- Assoc Prof. Jun Luo and I provided the initial research direction.
- Dr. Zhe Chen and I provided RF sensing training data samples.
- I designed framework and performed experiments.
- I provided experimental analysis.
- I wrote the drafts of the manuscript. The manuscript was revised together with Assoc Prof. Jun Luo, Dr. Zhe Chen, and Mr. Tianyue Zheng.

Chapter 5 is submitted to **Shuya Ding, Guanlin Li, Tianyue Zheng, and Jun Luo. "Adversarial Robustness is a Domain Knowledge: Fighting Data Scarcity in Transferring Robustness via Meta-Learning," Submitted to 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) for possible publication.**

The contributions of the co-authors are as follows:

- Assoc Prof. Jun Luo and I provided the initial research direction.
- I designed framework and performed experiments.
- Mr. Guanlin Li assisted in checking experimental settings.
- I provided experimental analysis.
- I wrote the drafts of the manuscript. The manuscript was revised together with Assoc Prof. Jun Luo, Mr. Guanlin Li, and Mr. Tianyue Zheng.

15 Jan 2021

.....
Date

.....
Shuya Ding

Abstract

Recent developments in computer vision using deep learning techniques have caused broad-ranging in all kinds of applications. However, the use of deep-driven computer vision techniques in improving system reliability is still a niche research direction. In this thesis, we present research efforts via deep-driven computer vision techniques for enhancing neural network based classification system reliability. Image Classification (IC) system and Radio-Frequency Based Human Activity Recognition (RF-HAR) system are selected as test-beds. The reliability enhancement is considered from two perspectives: i) enhance system robustness to defend against well-designed adversarial attacks, and ii) improve system performance in the face of extremely low data regime. Therefore, adversarial defense and knowledge transfer are the two main research objectives to be investigated in this thesis. To achieve these objectives, we consider the following approaches: i) an attack-agnostic adversarial defense framework to enhance the intrinsic robustness of IC system when it is attacked by adversarial examples, ii) a meta-learning based approach to transfer knowledge from seen environments, leading to adaptation of RF-HAR system in new environment with data scarcity, and iii) a meta-learning based adversarial defense framework to transfer robustness from robust IC system, giving rise to non-robust IC system robustness enhancement even under data scarcity.

To be specific, an attack-agnostic adversarial defense framework is first designed to improve the intrinsic robustness of IC system. Many researchers have proposed various adversarial attacks to generate adversarial examples. They can severely threaten the reliability of the system. Correspondingly, adversarial defense strategies are designed as countermeasures to improve system robustness. Although adversarial training is the most promising defense strategy by augmenting specific adversarial examples during training, it has limited classification system generalization capability and suffers from excessive

time complexity. An attack-agnostic defense framework named Feature Pyramid Decoder (FPD) is proposed to enhance the intrinsic robustness of the classification system. Meanwhile, it is without jeopardizing the ability to generalize clean samples. FPD could apply to all block-based systems. It integrates denoising layers and image restoration modules into a targeted system, and it also constrains the Lipschitz constant of the classification layer. Furthermore, a two-phase strategy is crafted for training the FPD-enhanced system. It utilizes ϵ -noisy neighborhood images with multi-task and self-supervised learning. Extensive experiments are conducted under a variety of well-known white-box attacks and black-box attacks on MNIST, the Street View House Numbers (SVHN), and CALTECH datasets. Results demonstrate that the FPD-enhanced system could improve intrinsic robustness against adversarial examples in an attack-agnostic manner. Besides, if adversarial training is conducted, the FPD-enhanced system performs better than their non-enhanced version.

Afterward, we propose a meta-learning based approach to transfer knowledge from seen environments, giving rise to the capability of RF-HAR system being adaptive to new environments with very few labeled data. To be specific, RF-HAR system rises as a promising solution for many applications. However, RF sensing (device-free) is often more sensitive to environmental changes than device-based sensing. Also, RF datasets strictly require on-line labeling during collection, starkly different from image and text data collections where human interpretations can be leveraged to perform off-line labeling. Therefore, existing solutions to RF-HAR system entail a laborious data collection process for adapting to new environments. To this end, RF-Net as a unified meta-learning based knowledge transfer framework for one-shot RF-HAR is proposed; it reduces the labeling efforts for environment adaptation to the minimum level. We innovate in two designs: i) a dual-path base HAR network, where both time and frequency domains are dedicated to learning powerful RF features including spatial and attention-based temporal ones, and ii) a metric-based meta-learning framework learning from seen environments, where HAR network is enabled to transform input into a representation suitable for classification via the similarity comparison and then environment adaptation can be performed with very few labeled data. It includes an RF-specific metric module along with a residual classification module. We conduct extensive experiments on three

representative RF sensing techniques, Wi-Fi radio, frequency-modulated continuous wave radio, and impulse radio, in multiple real-world indoor environments; all results strongly demonstrate the efficacy of RF-Net being adaptive to new environments with limited labeled data, compared with state-of-the-art baselines.

Last but not least, we further employ knowledge transfer to transfer robustness in the form of robust features, from adversarially-trained robust (source) IC system to a non-robust (target) IC system, and consequently to enhance robustness of target IC system even in extremely low data regime. We design a meta-learning based adversarial defense framework to perform this few-shot robustness transfer. As mentioned above, adversarial training is deemed as a potent defense strategy against powerful adversarial attacks. However, it requires sufficient well-designed adversarial examples, leading to an extra burden on crafting adversarial examples. One recent trend to handle this issue focuses on direct transferring adversarial robustness from a robust system to a non-robust system. However, the performance of these existing solutions may decay severely in low data regimes. Therefore, it motivates us to propose Meta Robustness Transfer (MRT) empowered by a non-parametric Robust Feature Selection Layer (RFSL) as a meta-learning based adversarial defense framework for fighting the data scarcity in transferring robustness. Guided by a robust source system, MRT essentially enables an adapted target system to learn an embedding function; this function embeds samples into robust features without involving adversarial training, and it enables a robust sample classification via a simple similarity comparison, avoiding the need for extensive labelled data. Consequently, the robustness of the target system can be enhanced. Through experiments in Mini-ImageNet, Caltech-UCSD Birds, and CIFAR-FS, we show that our approach can transfer robustness across tasks with different system architectures, different label spaces, and different datasets, in extremely low data regimes.

Acknowledgments

I want to start by thanking my parents, Ms. Liaosha Li and Mr. Huiran Ding, for providing me with a harmonious and warm environment to grow. My mother is an essential mentor who teaches me the importance of self-study and speculation. My father put my happiness as a top priority. Their hard work and dedication enable me to receive a high-quality education. I appreciate their never-failing adequate support.

When I decided to pursue a Ph.D. degree, I was fortunate to be welcomed by my supervisor, Prof. Jun Luo. I would like to express my sincere thanks and appreciation to him for all the efforts that he has taken in these years. He always encouraged me to broaden my horizon in research areas and never attacked my confidence. He was always patient to help me solve the difficulties throughout this journey. His insights and encouragement enabled me to have a better understanding of my research directions. Moreover, his high standard has also been helpful in enhancing the quality of my research work.

Early in my research, I also learned a lot from Prof. Hanwang Zhang. He took the time to point me promising research directions that inspired my works. I want to thank Prof. Hanwang Zhang for his imparting valuable knowledge to me. Moreover, I would also like to thank my thesis committee members, Prof. Sinno, Jialin Pan and Prof. Zhiping Lin, for their guidance and patience throughout this journey.

I am also extremely grateful to my colleagues at Energy Research Institute @ Nanyang Technological University (ERI@N), especially Dr. Liangmong Koh and Prof. Tianyang Zhao. They mentored me closely and encouraged me to adhere to complete the Ph.D. degree. My company ERI@N allows me to collect training samples and providing computational resources.

Last but not least, I would like to thank Fan Zhou for being my spiritual pillar in these years. Her optimism, care, and encouragement have been the key to healing my frail mental throughout these tough but memorable years.

Contents

Abstract	v
Acknowledgments	viii
List of Figures	xvi
List of Tables	xix
List of Abbreviations	xxi
List of Publications	xxiii
 1 Introduction	 1
1.1 Background	1
1.1.1 Adversarial Attacks	1
1.1.2 Data Scarcity	2
1.2 Thesis Statement	3
1.2.1 Adversarial Defense against Adversarial Attacks	4
1.2.2 Knowledge Transfer for Fighting with Data Scarcity	5
1.2.3 Knowledge Transfer for Coping with Both Adversarial Attacks and Data Scarcity	6
1.2.4 Objectives	7
1.3 Thesis Outline	8
 2 An Overview of Systems and Risks	 9
2.1 Image Classification (IC) System under Adversarial Attacks	9
2.1.1 Definition of Adversarial Attacks	9
2.1.2 Why Do Adversarial Examples Exist?	10
2.1.3 Adversarial Transferability	11
2.1.4 Poisoning & Evasion Attacks	11

2.1.5	White-Box & Black-Box Attacks	11
2.1.5.1	White-Box Attacks	12
2.1.5.2	Black-Box Attacks	14
2.2	Radio-Frequency Based Human Activity Recognition (RF-HAR) System Environment Adaptation under Data Scarcity	15
2.2.1	RF-HAR System	15
2.2.2	Modeling RF Sensing	17
2.2.2.1	Wi-Fi Radio	18
2.2.2.2	Frequency-Modulated Continuous Wave (FMCW) Radio	18
2.2.2.3	Impulse Radio (IR)	19
2.2.3	Environment Adaptation under Data Scarcity	21
3	Adversarial Defense via Enhancing System Intrinsic Robustness against Adversarial Attacks	22
3.1	Related Work	22
3.1.1	Gradient Masking	23
3.1.1.1	Shattered Gradients	23
3.1.1.2	Stochastic Gradients	23
3.1.1.3	Drawbacks	24
3.1.2	Randomization	24
3.1.2.1	Random Input Transformation & Feature Pruning	24
3.1.2.2	Random Noising	25
3.1.2.3	Drawbacks	25
3.1.3	Denoising	25
3.1.3.1	Pixel Level Denoising	26
3.1.3.2	Feature Level Denoising	27
	Non-Local Means	27
	Bilateral Filter	28
	Mean Filter and Median Filter	28
3.1.3.3	Drawbacks	28
3.1.4	Robust Optimization	29

3.1.4.1	Lipschitz Constant Regularization	29
3.1.4.2	Adversarial Training	30
3.1.4.3	Drawbacks	31
3.1.5	Our Proposal	32
3.2	Feature Pyramid Decoder	33
3.2.1	Front and Back Denoising Module	34
3.2.2	Image Restoration Module	35
3.2.3	Lipschitz Constant Constrained Classification	36
3.2.4	Training Strategy	38
3.3	Experiments	39
3.3.1	Exploration Experiments	40
	Inner Denoising Layers Implanted Positions Selection	40
	Denoising Approaches Selection	40
	Ablation Study	41
	Activation Functions Selection	42
	Inner Denoising Layers Selection	42
	Training Strategy Selection	42
	Comparison with the Related Work	42
3.3.2	Comparison Experiments	43
	Notation and Implementation Details	43
	On MNIST	43
	On SVHN	48
	On CALTECH-101 & CALTECH-256	49
	Summary	49
3.4	Chapter Summary	49
4	Knowledge Transfer via Meta-Learning against Data Scarcity	51
4.1	Related Work	52
4.1.1	Transfer Learning	52
	4.1.1.1 Model-Based	53
	4.1.1.2 Adversarial Learning-Based	53

4.1.1.3	Drawbacks	54
4.1.2	Meta-Learning	54
4.1.2.1	Optimization-Based	56
4.1.2.2	Metric-Based	57
4.1.2.3	Drawbacks	59
4.1.3	Our Proposal	59
4.2	RF-Net: One-Shot HAR	60
4.2.1	Problem Formulation	61
4.2.2	Meta Framework	62
4.2.2.1	RF Metric Module	63
4.2.2.2	Residual Classification Module	64
4.2.2.3	Training Strategy	64
4.2.3	Dual-Path Base Network	66
4.2.3.1	Spatial Module	67
4.2.3.2	Attention-Based Temporal Module	68
4.2.3.3	Classification Module	69
4.3	Experiments	69
4.3.1	Datasets	70
4.3.1.1	Wi-Fi	70
4.3.1.2	FMCW	71
4.3.1.3	IR	71
4.3.2	Baselines & Implementation	71
4.3.2.1	Baselines for Dual-Path Base Network	72
4.3.2.2	Baselines for RF-Net	72
4.3.2.3	Implementation	73
4.3.3	Hyperparameter Searching	73
4.3.3.1	Backbone Selection	74
4.3.3.2	Hidden Dimension Size & Activation Function Selection	75
4.3.4	Superiority of Proposed Network	76
4.3.4.1	Superiority of Dual-Path Base Network over Baselines	76
4.3.4.2	Superiority of RF-Net over Baselines	79

4.3.4.3	Wi-Fi vs. IR vs. FMCW	81
4.3.4.4	Impact of Number of Environments	82
4.3.5	Limitations	83
4.4	Chapter Summary	83
5	Robustness Knowledge Transfer via Meta-Learning against Both Data Scarcity and Adversarial Attacks	84
5.1	Related Work	85
5.1.1	Adversarial Transfer Learning	85
5.1.1.1	Drawbacks	86
5.1.2	Adversarial Meta-Learning	86
5.1.2.1	Drawbacks	88
5.1.3	Our Proposal	88
5.2	Meta Robustness Transfer (MRT)	89
5.2.1	Overview	89
5.2.2	Robust Feature Selection Layer	90
5.2.3	Adapted System via Meta-Learning	91
5.2.3.1	MRT Training Phase I	91
	C-Score Losses	92
	Robust Feature Selection	92
5.2.3.2	MRT Training Phase II	93
	Remark:	93
5.3	Experiments	94
5.3.1	Ablation Studies	95
5.3.2	Transfer across Different Architectures	95
	Experiment Setting	95
	Results	96
5.3.3	Transfer across Different Datasets	98
	Experiment Setting	98
	Results	99
5.3.4	Black-Box Adversarial Robustness	100
5.3.5	Overall Performance	101
5.4	Chapter Summary	104

6 Conclusion	105
A Mathematical Proofs in Chapter 3	109
A.1 Proof: Lipschitz Constant Constraint for Fully-Connected Network . . .	109
B Experimental Results in Chapter 5	116
B.1 Input Gradient Adversarial Matching Performance	116
B.2 Transfer Robustness across Systems and Datasets	117
B.3 Time Complexity	119
References	120

List of Figures

2.1	Deep fool.	13
2.2	Wi-Fi CSI heatmaps of two different activities: wiping the board and walking around. It is virtually impossible to intuitively recognize them by human eyes.	16
2.3	The RF signal matrix and corresponding heatmaps for three RF techniques in terms of the “walking” activity.	17
2.4	Environment influence: the same activity “walking” leads to distinct signal matrices and extracted features.	20
3.1	The structure of the block-based IC system, enhanced with the proposed framework named FPD: it consists of the Lipschitz constant constrained classification layer FPD^{LCC} ; the front denoising module FPD^{FD} , the image restoration module FPD^{R} , a middle denoising layer and the back denoising module FPD^{BD} . ϵ -neighbourhood noisy samples $\mathbf{x}^{\text{noisy}}$ and original samples $\mathbf{x}^{\text{clean}}$ are used to train the FPD. Orange, blue and green blocks represent the original components of the IC system, the proposed components that implanted to the IC system, the modified components of the IC system, respectively.	34
3.2	Three types of denoising layers which we have experimented on. (a) an inner denoising layer that linking two residual blocks with bottleneck, (b) an inner denoising layer that linking two residual blocks without bottleneck and (c) a middle denoising layer that denoising the input of the last part without bottleneck.	35

3.3	Implementation details of Lipschitz constant constrained (FPD ^{LCC}). It is implemented by involving a squeezing activation function to the output of a fully connected layer, i.e. Tanh.	36
3.4	Implementation details of two-phase training strategy utilizing self-supervised and multi-task learning: the enhanced IC system FPD, in which FPD ^R refers to the image restoration module; FPD ^{FD} stands for the front denoising module; FPD ^{BD} stands for the back denoising module; FPD ^{LCC} refers to the modified classification layer; $\mathbf{x}^{\text{noisy}}$ are the samples in the ϵ -neighbourhood of each image. The first phase training is optimized by $\mathcal{L}_2(\mathbf{x}^{\text{clean}}, \mathbf{x}'^{\text{clean}})$ loss. If $\mathcal{L}_2(\mathbf{x}^{\text{clean}}, \mathbf{x}'^{\text{clean}}) > \lambda$, only the parameters of FPD ^R and FPD ^{FD} is updated. Once the $\mathcal{L}_2$ loss reaches the λ , the cross entropy (CE) loss $\mathcal{L}^{\text{CE}}$ with $\mathcal{L}_2$ loss jointly trains the enhanced IC system. Then, the second phase train the enhanced IC system further, jointly optimized by $\mathcal{L}^{\text{CE}}$ and $\mathcal{L}_2$ loss.	37
3.5	Adversarial images (a) vs. the output of image restoration module from adversarial images (b) and clean images (c). Images are reproduced from the data in Table 3.6 (enhanced ResNet-101 attacked by PGD).	48
4.1	Overall design of RF-Net $\mathbf{q}_\Omega$	62
4.2	Overall design of dual-path base network $\mathbf{f}_\Phi$	66
4.3	Two alternative spatial modules.	67
4.4	Attention-based temporal module.	68
4.5	CNN backbone selection.	74
4.6	Hidden dimension size.	75
4.7	Activation function selection.	75
4.8	Overall comparison results for dual-path base network evaluation on Wi-Fi dataset.	76
4.9	Overall comparison results for dual-path base network evaluation on FMCW dataset.	77
4.10	Overall comparison results for dual-path base network evaluation on IR dataset.	77

4.11 FMCW attention map $\mathcal{A}$ of two different activities: walking and sitting down.	78
4.12 Overall comparison results for dual-path base network evaluation on Wi-Fi, FMCW, IR datasets in terms of general classification.	78
4.13 Overall comparison results for RF-Net evaluation on Wi-Fi dataset. . . .	79
4.14 Overall comparison results for RF-Net evaluation on FMCW dataset. . .	79
4.15 Overall comparison results for RF-Net evaluation on IR dataset.	79
4.16 Accuracy density plot of all evaluated networks.	80
4.17 Performance on RF sensing types.	81
4.18 Performance heatmap: accuracy improvement from SPATIAL-SEP, while a darker color denotes a higher accuracy improvement.	81
4.19 Impact of number of environments.	82
5.1 c-Score (y-axis) between each feature (x-axis) and the corresponding input gradient map computed on MNIST; the selected samples are all correctly classified. Note the remarkable similarity between (a) and (b) under an adversarially-trained system, which simply does not exist between (c) and (d) under a normally-trained system.	88
5.2 Overview of MRT framework: it transfers robustness from robust (source) system to data-scarce non-robust (target) system, driven by a two-phase meta-training strategy.	90
5.3 Overview of RFSL.	91
5.4 Time complexity comparison, given Resnet34 as f_θ on Mini-ImageNet. . .	98
5.5 Overall performance comparison. MI, CUB, and FS refer to Mini-ImageNet, CUB-200, and CIFAR-FS.	101
5.6 Qualitative results of Resnet34 as f_θ on Mini-ImageNet.	103
B.1 Time complexity comparison, given Resnet18 as f_θ on Mini-ImageNet. . .	119

List of Tables

3.1	Overall results of the exploration experiments with ResNet-101 on MNIST.	41
3.2	Robustness evaluation results (Accuracy %, Attack Time (min)) in thwarting the white-box attacks with ResNet-101 on MNIST.	43
3.3	l_2 Metrics: robustness evaluation results (Accuracy(%), Attack Time(min)) in thwarting the black-box attacks with ResNet-101 on MNIST. $\mathcal{F}$ -Average ($\mathcal{O}$ -Average) measure the average performance of $\mathcal{F}$, $\mathcal{F}^{\text{FGSM}}$, $\mathcal{F}^{\text{PGD}}$ ($\mathcal{O}$, $\mathcal{O}^{\text{FGSM}}$, $\mathcal{O}^{\text{PGD}}$).	44
3.4	l_∞ Metrics: robustness evaluation results (Accuracy(%), Attack Time(min)) in thwarting the black-box attacks with ResNet-101 on MNIST. $\mathcal{F}$ -Average ($\mathcal{O}$ -Average) measure the average performance of $\mathcal{F}$, $\mathcal{F}^{\text{FGSM}}$, $\mathcal{F}^{\text{PGD}}$ ($\mathcal{O}$, $\mathcal{O}^{\text{FGSM}}$, $\mathcal{O}^{\text{PGD}}$).	45
3.5	l_∞ Metrics: Robustness evaluation results (Accuracy %) in thwarting the black-box attacks with ResNet-101, ResNet-50 and ResNeXt-50 on SVHN.	46
3.6	Robustness evaluation results (Accuracy %) in thwarting the white-box attacks with ResNet-101, ResNet-50 and ResNeXt-50 on SVHN.	47
4.1	Time required to solve 10,000 miniImageNet episodes of 10 samples each.	59
4.2	Datasets information.	70
5.1	Accuracy (%) of ablation studies.	95
5.2	Accuracy (%) of Mobilenet as f_θ under 3-way 1-shot tasks on Mini-ImageNet.	96
5.3	Accuracy (%) of Mobilenet as f_θ under 3-way 5-shot tasks on Mini-ImageNet.	96
5.4	Accuracy (%) of Resnet34 as f_θ under 3-way 1-shot tasks on Mini-ImageNet.	97
5.5	Accuracy (%) of Resnet34 as f_θ under 3-way 5-shot tasks on Mini-ImageNet.	97

5.6	Accuracy (%) of defending against PGD-7 attacks with different perturbation budgets under 3-way 1-shot on Mini-ImageNet.	97
5.7	Accuracy (%) of Resnet18 as f_θ under 3-way 1-shot on CIFAR-FS.	99
5.8	Accuracy (%) of Resnet18 as f_θ under 3-way 5-shot on CIFAR-FS.	100
5.9	Accuracy (%) of Resnet18 as f_θ under 3-way 1-shot on CUB-200.	100
5.10	Accuracy (%) of Resnet18 as f_θ under 3-way 5-shot on CUB-200.	101
5.11	Accuracy (%) of Resnet18 as f_θ under 3-way 1-shot on Mini-ImageNet.	101
5.12	Accuracy (%) of Resnet18 as f_θ under 3-way 5-shot on Mini-ImageNet.	102
5.13	Accuracy (%) of defending against black-box attacks under 3-way 1-shot on Mini-ImageNet.	102
5.14	Accuracy (%) of defending against black-box attacks under 3-way 5-shot on Mini-ImageNet.	103
A.1	The structure of the network.	110
B.1	IGAM performance on clean samples. MI and FS are Mini-ImageNet and CIFAR-FS.	116
B.2	Accuracy (%) of Mobilenet as f_θ under 5-way 1-shot tasks on Mini-ImageNet.	117
B.3	Accuracy (%) of Mobilenet as f_θ under 5-way 5-shot tasks on Mini-ImageNet.	117
B.4	Accuracy (%) of Resnet34 as f_θ under 5-way 1-shot tasks on Mini-ImageNet.	117
B.5	Accuracy (%) of Resnet34 as f_θ under 5-way 5-shot tasks on Mini-ImageNet.	118
B.6	Accuracy (%) of Resnet18 as f_θ under 5-way 1-shot tasks on CIFAR-FS.	118
B.7	Accuracy (%) of Resnet18 as f_θ under 5-way 5-shot tasks on CIFAR-FS.	118
B.8	Accuracy (%) of Resnet18 as f_θ under 5-way 1-shot tasks on Mini-ImageNet.	118
B.9	Accuracy (%) of Resnet18 as f_θ under 5-way 5-shot tasks on Mini-ImageNet.	118

List of Abbreviations

DL	Deep Learning.
NN-IC	Neural Network-Based Image Classification.
RF-HAR	Radio-Frequency Based Human Activity Recognition.
FPD	Feature Pyramid Decoder.
HAR	Human Activity Recognition.
RFSL	Robust Feature Selection Layer.
MRT	Meta Robustness Transfer.
CNN	Convolutional Neural Network.
RNN	Recurrent Neural Network.
L-BFGS	Limited-Memory Broyden Fletcher Goldfarb Shanno.
FGSM	Fast Gradient Sign Method.
BIM	Basic Interactive Method.
PGD	Projected Gradient Descent.
C&W	Carlini & Wagner.
ZOO	Zeroth Order Optimization.
RF	Radio-Frequency.
RSSI	Received Signal Strength Indicator.
COTS	Commercial-Off-The-Shelf.
CSI	Channel State Information.
FMCW	Frequency-Modulated Continuous Wave.
OFDM	Orthogonal Frequency Division Multiplexing.
FFT	Fast Fourier Transform.
IR	Impulse Radio.
TV	Total Variance.
DP	Differential Privacy.
RSE	Random Self-Ensemble.
GAN	Generative Adversarial Networks.
HGD	High-Level Feature Guided Denoising.
DUNET	Denoising U-Net.
DAE	Denoising Auto-Encoder.
EAT	Ensemble Adversarial Training.
YOPO	You Only Propagate Once.
BPDA	Backward Pass Differentiable Approximation.

EOT	Expectation over Transformation.
MMD	Maximum Mean Discrepancy.
MAML	Model-Agnostic Meta-Learning.
PN	Prototypical Network.
IGAM	Input Gradient Adversarial Matching.
ADML	ADversarial Meta-Learner.
AQ	Adversarial Querying.
LCC	Lipschitz Constant Constraint.
FD	Front Denoising Module.
BD	Back Denoising Module.
MAC	Memory Access Cost.
NN	Neural Network.
FC	Fully Connected.
CE	Cross Entropy.
SVHN	Street View House Numbers.
tx-rx	Transmitter-Receiver.
FFT	Fast Fourier Transform.
RELU	Rectified Linear Unit.
LSTM	Long Short Term Memory.
MSE	Mean Squared Error.
KL	Kullback-Leibler.
c-Score	Correlation Score.
NT	Normally-Trained.
CUB-200	Caltech-UCSD Birds.

List of Publications

International Conferences

- Shuya Ding, Guanlin Li, Tianyue Zheng, and Jun Luo. "Adversarial Robustness is a Domain Knowledge: Fighting Data Scarcity in Transferring Robustness via Meta-Learning," Submitted to 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) for possible publication.
- Shuya Ding¹, Zhe Chen¹, Tianyue Zheng, and Jun Luo. "RF-Net: A Unified Meta-Learning Framework for RF-Enabled One-Shot Human Activity Recognition," in 2020 ACM Conference on Embedded Networked Sensor Systems (SenSys).
- Guanlin Li¹, Shuya Ding¹, Jun Luo, and Chang Liu. "Enhancing Intrinsic Adversarial Robustness via Feature Pyramid Decoder," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

International Journals

- Shuya Ding, Chaoyu Dong, Tianyang Zhao, Liangmong Koh, Xiaoyin Bai and Jun Luo, "A Meta-Learning Based Multi-Modal Neural Network for Multi-Step ahead Battery Thermal Runaway Forecasting," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2020.3015555.

¹These authors contributed equally to the work.

Chapter 1

Introduction

1.1 Background

Deep Learning (DL), as a subfield of machine learning, studies statistical models named deep neural networks to learn complex and hierarchical representations from raw data [1]. With the growth in data volume, computational, and storage capacities, the modern era of DL has started with the creation of more complex and deep architectures [2, 3]. DL has been deeply investigated in numerous applications to achieve artificial intelligence, such as image classification [3], human activity recognition [4], object detection [5], and semantic segmentation [6], etc. Despite this, DL has made a tremendous impact on computer vision, reaching previously unattainable performance on many tasks. The use of deep-driven computer vision techniques to improve the reliability of *the classification system built on neural network* is still a niche research direction. In this thesis, reliability enhancement of neural network-based classification system is regarded as *classification accuracy enhancement in the face of two types of risks: well-designed adversarial attacks and extremely low data regime*.

1.1.1 Adversarial Attacks

Neural network has become essential in many machine learning tasks and has been successful. Nowadays, Image Classification (IC) system, which classifies given input images into specific category labels, achieves almost human-level accuracy. Generally, IC system is a kind of Convolutional Neural Network (CNN). Thanks to its satisfying performance, it has been widely used in safety-critical tasks. For instance, IC has been exploited to

recognize road signs in autonomous vehicles [7]; they have also been involved in face recognition tasks for security surveillance [8]. In these safety-critical tasks, IC system is required to have convincing performance, i.e., high reliability.

Unfortunately, researchers have demonstrated that IC systems may be compromised by well-designed *adversarial attacks* [9]. They can manipulate model inputs (clean samples) to create *adversarial examples*, who subvert input-output mapping completely. Adversarial attacks are defined as follows: though easily distinguishable by human eyes, they are often wrongly classified by the IC system. In other application domains such as text and audio, adversarial attacks with similar meanings are proposed. For instance, [10] intends to perturb limited characters or words to fool text classification system. [11] manipulates edges to mislead graph neural networks.

Since the seminal work by [12, 13], many follow-up works have demonstrated a great variety of methods in generating adversarial samples. More specifically, most convolutional layers are very sensitive to perturbations brought by adversarial samples [14, 15], resulting in misclassifications. These so-called adversarial attacks may adopt either white-box attack or black-box attack, depending on the knowledge of the target network. White-box attacks assume that attackers can access all the information of the target IC system, such as network architectures, parameters, and gradients, etc. Black-box attacks assume that attackers can not access interior information of the IC system. These adversarial samples result in menace in IC system’s reliability. We will describe well-known adversarial attacks in Chapter 2

1.1.2 Data Scarcity

While neural network-based systems have been highly successful in data-intensive applications, their performance is often degraded when labeled training samples are insufficient. Although this performance degradation can be circumvented by collecting sufficient labeled training samples, the whole process is laborious and expensive. Consequently, data scarcity (limited supervised information in this thesis) may pose a challenge in landing neural network-based systems in the real world. Essentially, this degraded performance deviates from the ultimate goal of artificial intelligence, as human beings are capable of

learning new tasks rapidly with limited supervised information by utilizing what they learned in the past.

We take the Radio-Frequency Based Human Activity Recognition (RF-HAR) system as an example. Human Activity Recognition (HAR) has been attracting attention increasingly in both academia and industry, giving its promising potential in real-world applications such as smart home [16–18], health care [19–21], and fall detection [22–24]. RF sensing utilize Radio-Frequency (RF) signals [18, 25, 26] to provide device-free sensing. It offers enough resolution to perform HAR without infringing on personal privacy, and it is quite energy-efficient. Moreover, its performance is robust even under various temperature or lighting conditions. Therefore, RF-HAR is deemed as the most promising solution.

However, RF sensing is often more sensitive to environmental changes. Therefore, RF sensing requires a high volume of training data to re-train its system once the sensing environment is altered. Unfortunately, RF sensing data can not be recognized by human beings intuitively. Consequently, RF datasets need online labeling during collection, leading to an extra burden on data collection. Existing solutions to the RF-HAR system suffer from a laborious data collection process for adapting to new environments. A solution for RF-HAR system performing environment adaptation in data scarcity is necessary. More details of RF-HAR system and the challenge posed by data scarcity are described in Chapter 2

1.2 Thesis Statement

The focus of this thesis is on presenting research efforts via deep-driven computer vision techniques for enhancing reliability of neural network-based classification system in the face of the two risks - adversarial attacks and data scarcity. Adversarial defense and knowledge transfer are the main research directions to be investigated. The whole thesis design is carried out as follows. We first employ adversarial defense to improve the reliability of classification system in the face of adversarial attacks. The purpose of first work is to dig deeper into adversarial attacks and its countermeasure. Subsequently, we investigate the capability of knowledge transfer on solving second type of risk - data

scarcity. After researching the first two methods and these two types of risk, we are motivated to perform adversarial defenses with knowledge transfer to solve these two types of risks concurrently. Therefore, this idea brings us to the third work. Essentially, this thesis is carried out from both an implementation and a theoretical standpoint.

1.2.1 Adversarial Defense against Adversarial Attacks

To counter adversarial attacks, *adversarial defense* is an active research area. Many adversarial defense strategies have been proposed, such as gradient masking [27, 28], randomization [29, 30], denoising [31, 32], and robust optimization [33, 34], etc. Gradient masking approaches intend to mask gradients to defend against gradient-based adversarial attacks. Randomization based defense believes that IC system is robust to random perturbations. Therefore, this line of works intends to randomize adversarial perturbation into random perturbation. Denoising based approaches are much more straightforward; they directly employ denoising methods to alleviate the effect caused by adversarial perturbations on either pixel level or feature level. Last but not least, robust optimization improves IC system robustness by modifying learning manner.

However, most of these approaches either be defeated by more powerful attacks or suffer from high computational complexity caused by crafting specific adversarial examples, training additional networks, and complex optimization. Adversarial training [9, 34] is often deemed as the most powerful defense strategy by augmenting sufficient well-designed adversarial examples during training. Consequently, it leads to an extra burden on crafting adversarial examples, and it has limited classification system generalization capability. Little attention has been given to the direct design of robust frameworks in an *attack-agnostic* manner. In this thesis, attack-agnostic means that adversarial defense strategy can cope with most well-known adversarial attacks and without crafting any specific adversarial examples.

Therefore, in Chapter 3, we propose a novel attack-agnostic adversarial defense framework named Feature Pyramid Decoder (FPD) to enhance the intrinsic robustness of all types of block-based IC system. Meanwhile, it is without jeopardizing the ability to generalize clean samples. To be specific, FPD framework implants a target IC system with both denoising and image restoration modules to filter an input image at multiple

levels. We also deploy a Lipschitz Constant Constraint at the classification layer to limit the output variation in the face of attack perturbation. Moreover, a two-phase strategy is designed for strategically training the enhanced IC system. We validate our framework performance on MNIST, the Street View House Numbers (SVHN), and CALTECH datasets in defending against a variety of well-known white-box attacks and black-box attacks, achieving promising results. Moreover, under adversarial training, an enhanced IC system is much more robust than the non-enhanced version.

1.2.2 Knowledge Transfer for Fighting with Data Scarcity

Researchers are investigating knowledge transfer to imitate human behavior in learning knowledge from known tasks and transferring it to new tasks even with minimal supervised information to enhance RF-HAR system performance in the face of environment adaptation and data scarcity [25, 35, 36]. Related works either utilize transfer learning or meta-learning. *et al.* [35] employ parameter sharing based transfer learning to perform RF-HAR across different environments (deployment sites). They directly copy weights learned from seen environments (known tasks) and then fine-tune them with the limited labeled unseen environment (new task) data samples. Jiang *et al.* [25] utilize adversarial based transfer learning. They propose an adversarial RF-HAR network to eliminate environment-related information and extract environment invariant feature representation.

However, [25, 35] have one major drawback. They all focus on knowledge transfer to a single unseen environment. Therefore, the whole fine-tuning process has to be performed again with a substantial amount of labeled data when they are applied to a new environment. That means extra efforts in collecting data are still required. This drawback has been demonstrated in [26], and experimental results show that the performance of [25, 35] is degraded severely in a new environment if provided labeled data samples in the new environment are insufficient. To better tackle this challenge, we need to consider emerging alternatives, and meta-learning provides an alternative.

Recently, [36] first attempt to adopt optimization-based meta-learning to construct an adaptive wearable sensing system. Meta-learning endows the capability of learning how to adapt to multiple unseen environments by rehearsing related seen environments. Although it achieves promising performance in wearable sensing context, its

generalization ability is questionable in device-free RF sensing environments. As stated in [37], optimization-based meta-learning would suffer from generalization difficulty in high-dimensional parameter space. Compared with wearable sensing data, the dimension of RF sensing data is much higher. Consequently, a powerful feature extraction network (with a high-dimensional parameter space) is demanded, leading to optimization-based meta-learning approaches largely inapplicable.

We are motivated to utilize knowledge transfer for reducing the labeling efforts for RF-HAR system being adaptive to new environments to the minimum level. In Chapter 4, we propose RF-Net, a unified metric-based meta-learning based knowledge transfer framework for RF-enabled one-shot HAR, delivering the capability of being adaptive to new environments with very few labeled data. To be specific, we first innovatively design a dual-path activity recognition (base) network, learning influential spatial-temporal features from both time and frequency domains for enhancing HAR accuracy. Then, we propose a metric-based meta-learning framework to endow fast adaptation capability to the base network. It transfers knowledge from observed environments, transforming inputs into a representation suitable for classification through similarity comparison and then allowing environment adaptation with minimal labeled data. We conduct extensive experiments on representative RF sensing techniques, Wi-Fi radio, frequency-modulated continuous wave radio, and impulse radio, in many indoor environments. Results demonstrate the superior performance of our proposed RF-Net on performing environment adaptation with limited labeled data, compared against multiple baselines.

1.2.3 Knowledge Transfer for Coping with Both Adversarial Attacks and Data Scarcity

Throughout the first two methods, we are motivated by the idea that we can overcome these two types of threats simultaneously by incorporating adversarial defense with knowledge transfer. Recall that knowledge transfer based learning systems can transfer knowledge from known tasks to new tasks. Inspired by it, we are possible to cast “system robustness” as a type of knowledge and transfer it from a robust adversarially-trained (source) IC system to a non-robust (target) IC system. Consequently, it can enhance the robustness of the target IC system in the face of adversarial attacks. It may give rise

to direct acquiring adversarial robustness from the adversarially-trained system, avoiding re-conducting adversarial training. Moreover, as mentioned in Sec. 1.2.2, knowledge transfer endows the potential of fighting with data scarcity as well.

Recently, there have been efforts in employing this idea to defend against adversarial attacks. [38] transfer the robust feature extractor weights from adversarially-trained source system to a target system and [39] transfers input gradient map so that a non-robust system can learn how robust system perceives pixel importance. Although these approaches are effective in transferring robustness, they fail to consider data scarcity in the target system and demand sufficient labeled data in the training target system. As stated in [38], the performance of transferring robustness would decay severely in low data regimes. Therefore, we should consider robustness transfer in the face of data scarcity.

In Chapter 5, we propose a meta-learning based adversarial defense framework to perform a few-shot robustness transfer in the form of robust features. We design a non-parametric Robust Feature Selection Layer (RFSL) to characterize the distribution of robust features. Empowered by RFSL, we specifically design Meta Robustness Transfer (MRT) as a meta-learning based training strategy to transfer the robust feature embedding from a robust IC system to a non-robust IC system. Our MRT learns to embed samples into robust features, reducing sample classification to straightforward similarity comparison and allowing robustness to be transferred with very few labeled data. Consequently, MRT can enhance the robustness of the target system without re-conducting adversarial training. Through extensive comparisons in Mini-ImageNet, Caltech-UCSD Birds, and CIFAR-FS, we demonstrate the efficacy of our approach in robustness transfer across different attacks with different IC system architectures, different label spaces, and different datasets, under data scarcity.

1.2.4 Objectives

Consequently, we formulate our thesis objectives: *We investigate adversarial defense and knowledge transfer, and propose three approaches to achieve following system reliability enhancement objectives: i) utilizing adversarial defense to enhance the intrinsic robustness of IC system when it is attacked by adversarial examples, ii) employing knowledge*

transfer to improve RF-HAR system environment adaptation performance in low data regime, and iii) performing adversarial defense with knowledge transfer to enhance IC system robustness in defending against adversarial examples, even in data scarcity.

1.3 Thesis Outline

Following the above thesis objectives, this dissertation makes the following contributions, each presented in an individual chapter. In Chapter 3, we propose a novel attack-agnostic adversarial defense framework to enhance the intrinsic robustness of all types of block-based IC system. In Chapter 4, we propose a unified meta-learning based knowledge transfer framework for RF-enabled one-shot HAR, delivering the capability of being adaptive to new environments with very few labeled data. In Chapter 5, we propose a meta-learning based adversarial defense framework to perform a few-shot robustness transfer in the form of robust features. Before diving into thesis contributions, we introduce the threat of the considered two risks to IC system and RF-HAR system in Chapter 2. Finally, the conclusion is drawn in Chapter 6.

Chapter 2

An Overview of Systems and Risks

In this chapter, we introduce the threat of the two types of risks: adversarial attacks and data scarcity to Image Classification (IC) system and Radio-Frequency Based Human Activity Recognition (RF-HAR) system.

2.1 Image Classification (IC) System under Adversarial Attacks

2.1.1 Definition of Adversarial Attacks

Researchers have started to pay more attention to investigating the weakness of neural networks, especially IC systems. Since the seminal work by [12, 13], many follow-up works have demonstrated a great variety of methods in generating *adversarial examples*: though easily distinguishable by human eyes, they are often misclassified by IC systems and called *adversarial attacks*.

Given a clean image $\mathbf{x}^{\text{clean}}$ which is correctly classified as class y by a IC system f_{θ} parameterized by θ , adversarial example $\mathbf{x}^{\text{adv}}$ is crafted so that it is similar to $\mathbf{x}^{\text{clean}}$ but misclassified by f_{θ} , and it is defined as follows:

$$\|\mathbf{x}^{\text{clean}} - \mathbf{x}^{\text{adv}}\|_p < \epsilon, \quad \text{s.t.} \quad f_{\theta}(\mathbf{x}^{\text{adv}}) \neq y, \quad (2.1)$$

where $\|\cdot\|_p$ denotes p -norm and ϵ represents the perturbation threshold imposed upon adversarial attacks. Eqn. (2.1) is also called *non-target attack* since attack mechanism focus on $f_{\theta}(\mathbf{x}^{\text{adv}}) \neq y$. If attackers intend to craft $\mathbf{x}^{\text{adv}}$ such that $f_{\theta}(\mathbf{x}^{\text{adv}}) = t \neq y$ where t is a pre-defined label, then this attack is so-called *target attack*.

2.1.2 Why Do Adversarial Examples Exist?

Szegedy *et al.* [12] posit that the existence of adversarial examples is caused by the high complexity of IC system. They argue that this high complexity leads to bad generalization in the low probability space of data. However, Goodfellow *et al.* [9] find that a simple linear system can be defeated by adversarial examples as well. Moreover, Madry *et al.* [34] also demonstrate that higher IC system capacity (more complex) would enhance system robustness.

Some researchers explain the existence from a decision boundary perspective. They find that adversarial examples are always close to the decision boundary of IC system and argue that the existence of adversarial examples may be the fact that the decision boundary is either too float [40], too curved [41], or inflexible [42]. Recent works hypothesize the inevitability of adversarial examples. Shafahi *et al.* [43] states that class-wise data distribution is not concentrated, leading to spaces for adversarial examples.

[44] regards adversarial examples as a consequence of IC system's sensitivity to well-generalizing features in the data. They state that the training objective of IC system is to maximize system accuracy. Therefore, the system intends to exploit any features to achieve this objective. However, there are certain features that are highly predictive yet imperceptible. They posit that these imperceptible features might be non-robust and be exploited to craft adversarial examples. They categorize features into three types:

- ρ -useful features: given $(\mathbf{x}, y) \sim \mathcal{D}$, features are called ρ -useful if they are correlated with ground truth label y in expectation and the expectation exceed threshold ρ . ρ -useful features can be represented as follows:

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [y \cdot h(\mathbf{x})] \geq \rho, \quad (2.2)$$

where h denotes feature extractor.

- β -robustly useful features: given adversarial perturbation δ , features remain β -useful. Formally, they are defined as follows:

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [y \cdot h(\mathbf{x} + \delta)] \geq \beta. \quad (2.3)$$

- Useful but non-robust features: features are regarded as useful but non-robust if they are ρ -useful but not β -robust features. These features may help improve classification performance but hinder accuracy in the face of adversarial examples.

2.1.3 Adversarial Transferability

An interesting property of adversarial examples is *adversarial transferability*. It is the phenomenon that if $\mathbf{x}^{\text{adv}}$ can attack the victim IC system, $\mathbf{x}^{\text{adv}}$ has a high probability of attacking other systems [45, 46]. Several works explain this phenomenon. [47] investigates orthogonal vectors named *adversarial subspaces*. Their linear combinations are adversarial perturbations. They state that adversarial transferability is the consequence of overlap in the adversarial subspaces between different IC systems. [44] explain adversarial transferability in another perspective. They posit that IC systems learn to exploit useful but non-robust features, which help improve classification performance in the face of clean samples but may hinder accuracy attacked by adversarial examples. It leads to adversarial examples that exploit this dependence. Since any IC system can learn similar non-robust features, adversarial examples exploiting such non-robust features will apply to these IC systems, leading to adversarial transferability.

2.1.4 Poisoning & Evasion Attacks

In various situations, attackers may have different adversaries' goals. Based on it, adversarial attacks can be categorized into *poisoning attacks* and *evasion attacks*. Poisoning attacks [48] assume that attackers are allowed to add adversarial examples into the system training dataset. In other words, attackers can access to training database. Poisoning attacks generally occur when web-based repositories are used. As web-based repositories collect malware samples for training, it provides space for attackers to poison the database. If attackers are not granted access to the database, they generally conduct evasion attacks [49]. In evasion attacks, attackers can only craft adversarial examples to evade detection or be misclassified by IC system. They are launched after IC system is trained.

2.1.5 White-Box & Black-Box Attacks

Evasion adversarial attacks can be further categorized based on adversaries' knowledge on what information can be accessed to the attacker, which is *white-box attacks* and *black-box attacks*.

2.1.5.1 White-Box Attacks

White-box attacks assume that attackers can access all the information of the target IC system, such as network architectures, parameters, and gradients, etc. Due to the entire disclosure of this information, reliability against white-box attacks is worth owning.

Biggio *et al.* [49] first exploit a discriminant function $g_\omega(\mathbf{x}^{\text{clean}})$ parameterized by ω to craft adversarial examples. Given $\mathbf{x}^{\text{clean}}$ correctly classified as y , if $g_\omega(\mathbf{x}^{\text{clean}}) \geq 0$, it marks $\mathbf{x}$ to be in class y and vice versa. Afterwards, they craft $\mathbf{x}^{\text{adv}}$ to minimize $g_\omega(\mathbf{x}^{\text{adv}})$ while constraining $\|\mathbf{x}^{\text{adv}} - \mathbf{x}^{\text{clean}}\|_1$ small. Once $g_\omega(\mathbf{x}^{\text{adv}})$ is negative, $\mathbf{x}^{\text{adv}}$ is classified not in class y and still similar to $\mathbf{x}^{\text{clean}}$, i.e., attack success.

Szegedy *et al.* [12] propose an attack by exploiting Limited-Memory Broyden Fletcher Goldfarb Shanno (L-BFGS) algorithm [50], the first target attack on IC system. They aim to search for minimal distorted $\mathbf{x}^{\text{adv}}$ as follows:

$$\min \quad \epsilon \|\mathbf{x}^{\text{clean}} - \mathbf{x}^{\text{adv}}\|_2^2 + \mathcal{L}(\theta; \mathbf{x}^{\text{adv}}; t) \quad (2.4)$$

$$\text{subject to } \mathbf{x}^{\text{adv}} \in [0, 1]^{N^{\text{pixel}}}, \quad (2.5)$$

where N^{pixel} denotes the number of pixels, the first objective term aims to find the minimal distorted $\mathbf{x}^{\text{adv}}$ measured by l_2 norm, and the second objective term intends to find $\mathbf{x}^{\text{adv}}$ with small loss $\mathcal{L}$ so that $f_\theta(\mathbf{x}^{\text{adv}}) = t \neq y$.

To improve attack efficiency, Goodfellow *et al.* propose one-step adversarial attack named Fast Gradient Sign Method (FGSM) [9]. FGSM mainly exploits gradient information to attack f_θ as follows:

$$\mathbf{x}^{\text{adv}} = \mathbf{x}^{\text{clean}} + \epsilon \text{sign}(\nabla_x \mathcal{L}(\theta; \mathbf{x}^{\text{clean}}; y)), \quad (2.6)$$

$$\mathbf{x}^{\text{adv}} = \mathbf{x}^{\text{clean}} - \epsilon \text{sign}(\nabla_x \mathcal{L}(\theta; \mathbf{x}^{\text{clean}}; t)), \quad (2.7)$$

where Eqn. (2.6) is non-target attack and Eqn. (2.7) is target attack.

As shown in Eqn. (2.6) and Eqn. (2.7), FGSM employs $\text{sign}(\nabla_x \mathcal{L})$ to guarantee same variation direction as gradient. Subsequently, the computed $\mathbf{x}^{\text{adv}}$ is with higher loss. Essentially, Eqn. (2.7) can be regarded as a one-step gradient descent to solve the following

Figure 2.1: Deep fool.

objective function:

$$\min \mathcal{L}(\theta; \mathbf{x}^{\text{adv}}; t) \quad (2.8)$$

$$\text{subject to } \|\mathbf{x}^{\text{clean}} - \mathbf{x}^{\text{adv}}\|_{\infty} < \epsilon \quad \text{and} \quad \mathbf{x}^{\text{adv}} \in [0, 1]^{N^{\text{pixel}}}. \quad (2.9)$$

Compared with iterative attacks such as L-BFGS attack, FGSM can generate adversarial examples with only one back-propagation step to boost attack speed. Afterward, to further improve attack ability, an iterative version of FGSM, Basic Interactive Method (BIM) is proposed by Kurakin *et al.* [51]. If random initialization is involved in BIM, it is also known as Projected Gradient Descent (PGD) [34]. Formally, $\mathbf{x}^{\text{adv}}$ is crafted by iterative conducting the following procedures:

$$\mathbf{x}_0 = \mathbf{x}, \quad (2.10)$$

$$\mathbf{x}_{t+1} = \text{Clip}_{\mathbf{x}, \epsilon}(\mathbf{x}_t + \nu \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\theta; \mathbf{x}_t; y))), \quad (2.11)$$

where Clip project argument to the surface of $\mathbf{x}$'s ϵ -neighbor ball and ν means the step size.

Moosavi-Dezfooli *et al.* [52] propose deep fool attack to craft $\mathbf{x}^{\text{adv}}$ by investigating decision boundary of f_{θ} around $\mathbf{x}^{\text{clean}}$. They aim to search a path which $\mathbf{x}^{\text{clean}}$ can be modified across the decision boundary hyperplane γ , and consequently mislead f_{θ} . As shown in Figure 2.1, given $(\mathbf{x}^{\text{clean}}, y)$, deep fool attack linearizes decision boundary hyperplane γ_3 using Taylor expansion, and consequently computes the orthogonal vector from $\mathbf{x}^{\text{clean}}$ to γ_3 . Afterwards, attacker could move along this vector to find $\mathbf{x}^{\text{adv}}$ such

that $f_\theta(\mathbf{x}^{\text{adv}}) = t$. The impressive attack performance also demonstrate that most test samples are closed to decision boundaries.

Carlini & Wagner’s attack (C&W) [53] intends to find the minimally distorted perturbation as follows:

$$\min \quad \|\mathbf{x}^{\text{clean}} - \mathbf{x}^{\text{adv}}\|_2^2 + \lambda(\max_{i \neq t} U(\mathbf{x}^{\text{adv}})_i - U(\mathbf{x}^{\text{adv}})_t)^+ \quad (2.12)$$

where λ stands for constant and U denotes logit function. Minimizing $(\max_{i \neq t} U(\mathbf{x}^{\text{adv}})_i - U(\mathbf{x}^{\text{adv}})_t)^+$ intends to find $\mathbf{x}^{\text{adv}}$ whose score for target class $t \neq y$ is larger than other classes.

Adversarial attack and defense confront each other. Finding a provable strongest attack, theoretical minimally distorted adversarial examples, is important for evaluating both attack and defense performance. Consequently, ground truth attack is proposed by [54]. They involve IC parameters θ and labeled clean sample $(\mathbf{x}^{\text{clean}}, y)$ as the constraints of linear programming optimization function. To be specific, they use $N_\iota(\mathbf{x}^{\text{clean}})$ to represent neighbors of $\mathbf{x}^{\text{clean}}$ within radius ι that at least one adversarial example $\mathbf{x}^{\text{adv}}$ exists. Ground truth attack keeps reducing ι until there is no $\mathbf{x}^{\text{adv}}$ in $N_\iota(\mathbf{x}^{\text{clean}})$. Consequently, the last founded $\mathbf{x}^{\text{adv}}$ is the ground truth adversarial example.

2.1.5.2 Black-Box Attacks

Black-box attacks assume that attackers can not access interior information of IC system f_θ , such as network parameters and network structures, etc. The only permission is to feed images and query output of f_θ . Even worse, some settings restrict query times. Generally, attackers can continuously feed crafted samples and then monitor output changes in response to input change. They exploit the input-output relationship to craft adversarial examples. The black-box setting is more practical and worth being investigated.

Papernot *et al.* [55] first propose black-box attack. They assume that: i) attackers can not access to θ and training database, ii) attackers can know the data domain, such as birds, photographs, facial images, and handwritten digits, etc, and 3) attackers can know the architecture of f_θ . Under these assumptions, Papernot *et al.* exploit transferability property of adversarial examples to attack f_θ . They keep feeding synthesized images $\hat{\mathbf{x}}$ to

f_θ and then collect corresponding $\hat{y}$. Afterwards, $\{(\hat{\mathbf{x}}, \hat{y})\}$ are utilized to train a substitute system $f_{\theta'}$, mimicking target IC system f_θ . Generally, similar architecture to that of f_θ would be used. Thanks to the fully exposed information of $f_{\theta'}$, aforementioned white-box attacks can be utilized to attack $f_{\theta'}$. Consequently, the crafted adversarial examples $\mathbf{x}^{\text{adv}}$ are further exploited to mislead target f_θ . Dong *et al.* [56] find that FGSM, PGD, and momentum-based iterative attacks generally have high transferability and shall be used for transferability based black-box attack.

Another type of black-box attack is named Zeroth Order Optimization (ZOO) [57]. They assume that attackers can access score information from f_θ . ZOO first computes the gradient information of $\mathbf{x}^{\text{clean}}$ by generating the variation of the score as the response of the changes of each pixel of the clean sample. Afterward, given the scraped gradient information, gradient-based white-box attacks could be exploited to attack f_θ .

2.2 Radio-Frequency Based Human Activity Recognition (RF-HAR) System Environment Adaptation under Data Scarcity

2.2.1 RF-HAR System

Recently, Human Activity Recognition (HAR) has been attracting attention increasingly in both academia and industry, giving its promising potential in real-world applications such as smart home [16–18], health care [19–21], and fall detection [22–24]. Generally, two categories for HAR have been explored: *device-based* (or wearable) and *device-free* (or contactless). Device-based HAR leverages the wearable devices such as smart phones or watches to recognize human activities [20, 58–60]. However, it may cause discomfort and extra burden, which leads to the alternative method of device-free HAR. This later method exploits camera-based image data [61–63], acoustic/ultrasonic signals [64–66], and Radio-Frequency (RF) signals [18, 25, 26] to achieve HAR.

Research efforts leveraging the efficacy of deep learning have been intensively carried out on camera-based approaches [62, 63, 67], but exploiting image data bears a risk of privacy infringement, especially in private places [16]. Meanwhile, acoustic/ultrasonic sensing often requires bulky devices and incurs a high energy consumption, hence rarely

Figure 2.2: Wi-Fi CSI heatmaps of two different activities: wiping the board and walking around. It is virtually impossible to intuitively recognize them by human eyes.

adopted in practice even though the performance can be excellent in noise-free environments [64, 65]. Fortunately, RF sensing offers just enough resolution to perform HAR without infringing personal privacy, and it is far more energy-efficient than sound, as RF signals are not generated by mechanical vibration. Moreover, its performance is robust even under various temperature or lighting conditions. Therefore, RF-HAR is deemed as the most promising solution, where Wi-Fi is often adopted.

Early RF-HAR exploited RSSI (Received Signal Strength Indicator) to analyze human activities [68, 69], collected by COTS (Commercial-Off-The-Shelf) Wi-Fi cards. However, RSSI can only represent coarse information of Wi-Fi signals, rather than fine-grained multipath effects generated by human activities. Therefore, many efforts have been recently devoted to extracting comprehensive information in Wi-Fi CSI (Channel State Informa-

Figure 2.3: The RF signal matrix and corresponding heatmaps for three RF techniques in terms of the “walking” activity.

tion) from Intel 5300 [70] Wi-Fi cards [25, 71–73]. As illustrated in Figure 2.2, body movements of different activities bring variations to CSI, so features generated from CSI could be leveraged for HAR. Whereas Wi-Fi sensing exploits devices originally designed for wireless communication purposes (hence inherently limited in sensing performance), dedicated RF sensing techniques are gaining momentum very recently. These techniques are typically supported by two types of radios, namely Frequency-Modulated Continuous Wave (FMCW) radio [74, 75] and impulse radio (IR) [76–78].

2.2.2 Modeling RF Sensing

Generally, all RF sensing approaches explore RF CSI to differentiate human activities. Therefore, we model the RF channel first, then examine three typical RF signals. According to [79], considering a pair of transmitter and receiver in an indoor environment

with P propagation paths, we have the following baseband RF channel model given a carrier frequency f_c :

$$h(t) = \sum_{p=1}^P \alpha_p e^{j2\pi f_c \tau_p} + n(t), \quad (2.13)$$

where α_p is the amplitude of p -th path, and $n(t)$ is Gaussian noise. Moreover, $\tau_p = \tau_p^S + \tau_p^D$ where τ_p^S and τ_p^D are the p -th time delays caused by static reflections and motion reflections, respectively. For a transmitted signal $s(t)$, the received signal becomes $y(t) = (h * s)(t)$, where $*$ denotes convolution. In the following, we stick to only one path (thus removing the subscript p), and we omit noise term for brevity.

2.2.2.1 Wi-Fi Radio

Wi-Fi communications utilize Orthogonal Frequency Division Multiplexing (OFDM) to encoding digital data on multiple subcarriers. Let L denote the number of OFDM subcarriers, s_ℓ denotes the ℓ -th subcarrier, then the received signal y_ℓ^W can be represented as follows:

$$y_\ell^W = \alpha_\ell e^{j2\pi f_\ell \tau} s_\ell, \quad \ell \in \{1, \dots, L\}, \quad (2.14)$$

where α_ℓ and f_ℓ are the amplitude and frequency of the ℓ -th subcarrier, respectively. Here the time variable t disappears because the bandwidth of Wi-Fi is so narrow that it is approximately deemed as time invariant. The channel state can then be estimated as $\hat{h}_\ell = y_\ell^W / s_\ell$, whose phase $\angle \hat{h}_\ell$ contains temporal feature τ .

Given a total K received packets, each of them offers a CSI vector $[\hat{h}_\ell]_{\ell \in \{1, \dots, L\}}$. Combining all these data, we obtain the ensemble input sample as a $K \times L$ signal matrix. Here we term the row index as *slow-time* and the column index as *fast-time*, as they represent sampling at different temporal scales. Figure 2.3a illustrates this signal model, while Figure 2.3b provides an example for a Wi-Fi matrix.

2.2.2.2 Frequency-Modulated Continuous Wave (FMCW) Radio

FMCW is a special type of radar implementation. Different from normal radios, FMCW radio transmits analog signals modulated in a continuously increasing frequency across

a wide bandwidth. Consequently, it sweeps across time with a fine-grained (frequency) resolution, equivalently convertible to a fine-grained range resolution. For FMCW radio, we denote the bandwidth by B and sweeping time span by T^S . According to [80], the received FMCW signal $y^F(t)$ is:

$$y^F(t) = \alpha \Pi(t - \tau) e^{-j2\pi(\beta\tau t + f_c\tau - 0.5\beta\tau^2)}, \quad (2.15)$$

where $\beta = B/T^S$ and $\Pi(t)$ is a rectangle function ranged from $-T^S/2$ to $T^S/2$.

We term a signal represented by Eq (2.15) a *frame*; it uses a frequency $\beta\tau$ and a phase $f_c\tau - \beta\tau^2/2$ to represent how the signal changes over time. Therefore, we perform an L -point Fast Fourier Transform (FFT) on each frame and get a set of L frequency components, which correspond to fast-time samples. Then we combine all K frames together to construct a $K \times L$ signal matrix (similar to that of Wi-Fi), as shown in Figure 2.3c.

2.2.2.3 Impulse Radio (IR)

Whereas FMCW radio uses a varying frequency to sweep time, IR transmits a pulse signal with an extremely short time duration. This enables IR signals to occupy a wide bandwidth B , again leading to a fine-grained time resolution. The received pulse signal $y^I(t)$ is:

$$y^I(t) = \alpha e^{j2\pi f_c\tau} e^{-0.5(t-0.5T_{tx}-\tau)^2 \epsilon_{tx}^{-2}}, \quad (2.16)$$

where $T_{tx} = \frac{1}{B}$ is the signal duration, and $\epsilon_{tx} = \frac{1}{2\pi B_{-10 \text{ dB}}(\log_{10}(e))^{1/2}}$ is the standard deviation determining the -10 dB bandwidth.

Similar to FMCW, if we consider L samples of a pulse as a *frame*, combining K such frames again gives us a $K \times L$ signal matrix, where the fast-time directly corresponds to the temporal sample indices of a frame, as shown in Figure 2.3d.

Remark: If we deem this matrix as an image (common input to deep learning systems such as CNN [81, 82]), we may stack multiple such matrices derived from different transmitter-receiver (tx-rx) pairs together so that each tx-rx pair corresponds to an input channel. Although these three RF sensing techniques share a similar model, they offer

Figure 2.4: Environment influence: the same activity “walking” leads to distinct signal matrices and extracted features.

very different time (thus range) resolutions: the 20 MHz bandwidth of Wi-Fi can only translate to a range resolution of about 15 m, but both FMCW and impulse radios, with more than 1 GHz bandwidth can achieve a centimeter-level resolution instead.

Existing RF-sensing solutions mostly leverage machine learning techniques to extract features and classify activities [71, 72, 74, 83–87]. Although their adopted RF signals and designed algorithms vary, their input data can all be unified under the RF signal matrix as subcategories. For example, the proposals in [71, 72, 84–86, 88] extract Doppler shift only along the slow time axis in Figure 2.3b, in order to capture the velocities of moving parts of a target and in turn to perform classifications. Other proposals [74, 87] utilize multiple tx-rx pairs to extract range information embedded along the fast time dimension in Figure 2.3c, so as to predict human poses. Essentially, existing solutions only utilize partial information provided by RF signals. So far, there is no unified framework to handle all three RF sensing techniques.

2.2.3 Environment Adaptation under Data Scarcity

Although RF-sensing solutions mostly leverage machine learning techniques to extract more features, exploiting more features is a double-edged sword: it improves activity classification under noise-free circumstances, but it also makes RF sensing more sensitive to environmental changes in practice. Figure 2.4 demonstrates the influence of this sensitivity. Therefore, similar to other device-free sensing techniques [63, 65, 89], RF sensing requires a high volume of training data to re-train its system if the sensing environment is altered (e.g., changing the radio locations and/or the room furniture layout/ different subjects). This weakness is inherent to device-free sensing as, unlike device-based sensing, it senses both the subject and the background.

Unfortunately, as illustrated in Figure 2.2, human beings cannot intuitively recognize different activities from RF sensing data (as opposed to images). Consequently, human interpretation cannot be exploited for conducting off-line labeling, which is essential for image and text data collections. As a result, RF sensing has to endure a much more laborious process (compared with computer vision and natural language processing) to gather labeled data sufficiently in order for RF-HAR approaches to be able to adapt to new environments with satisfactory performance. Therefore, a solution for RF-HAR system performing environment adaption in data scarcity is essential.

Chapter 3

Adversarial Defense via Enhancing System Intrinsic Robustness against Adversarial Attacks¹

As described in Sec. 2.1, the Image Classification (IC) system can be defeated by well-designed adversarial attacks, imperceptibly perturbed clean image that produces incorrect predictions in IC system. To defend against these adversarial attacks, many adversarial defense strategies have been proposed. However, the direct design of robust frameworks in an attack-agnostic manner is rarely studied. In this chapter, we give attention to this direction, and we propose a novel attack-agnostic adversarial defense framework named Feature Pyramid Decoder (FPD), applicable to enhance intrinsic robustness of all types of block-based IC systems. This chapter is organized as follows. We first provide an overview of existing adversarial defense strategies. Afterward, we describe each component of our proposed FPD and two-phase training strategy. Then, we validate the efficacy of FPD on MNIST, the Street View House Numbers (SVHN), and CALTECH datasets in defending against a variety of well-known white-box and black-box attacks.

3.1 Related Work

In order to improve IC system reliability in defending against adversarial examples, many defense algorithms are proposed as countermeasures. These algorithms can be categorized

¹The work in this chapter has been published in [90]

as follows: gradient masking, randomization, denoising, and robust optimization, etc.

3.1.1 Gradient Masking

Gradient masking focuses on obfuscating gradients to defend against gradient-based adversarial approaches such as Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and Basic Interactive Method (BIM), etc.

3.1.1.1 Shattered Gradients

Some works intend to shatter gradients. They train IC system after pre-processing input data x^{clean} with certain non-differentiable processor μ . Consequently, IC system $f_{\theta}(\mu)$ is non-differentiable and it can defend against gradient-based adversarial attacks. For instance, Buckman *et al.* [27] propose thermometer encoding. It discretizes pixel value into a l -dimensional binary vector. Image information is stored in binary vectors.

Guo *et al.* [28] investigate various image processors to block up the smooth connection between input sample and its corresponding IC system output, such as image cropping, image re-scaling, bit-depth reduction [91], image quilting [92], and image compression [93]. To be specific, image cropping and re-scaling intend to modify the spatial positioning of the adversarial perturbation, which is essential in crafting adversarial attacks. Bit-depth reduction and image compression perform quantization to remove small adversarial variation. Image quilting defenses assume that a clean patches database is provided. It first breaks the input image into small patches, and then it finds the most similar patches from the database for each image patch. Subsequently, all selected clean patches are stitched together as a synthesized image without involving any pixels crafted by the attacker.

3.1.1.2 Stochastic Gradients

Another line of approaches randomizes gradients by randomizing the prediction system. They first train a set of IC systems $\{f_{\theta_0}, f_{\theta_1}, \dots, f_{\theta_M}\}$ by either randomly dropping neurons of some layers [94], or resize input images to random size [29]. Afterward, they randomly choose one trained IC system during each evaluation. Therefore, attackers cannot exploit gradient information that scraped from f_{θ_i} to attack selected f_{θ_j} .

3.1.1.3 Drawbacks

Gradient masking mainly focuses on confounding the adversaries, but it cannot erase the existence of adversarial examples. Most gradient masking approaches such as shatter gradients and stochastic gradients can be broken by Athaly *et al.* [95] in the white-box setting. To be specific, Backward Pass Differentiable Approximation (BPDA) is designed to defeat the aforementioned shattered gradient approaches. Essentially, they replace non-differentiable processor μ with a similar differentiable processor $\tilde{\mu} \sim \mu$ during backward pass. Therefore, the gradient flow is rebuilt. A method named Expectation over Transformation (EOT) is proposed to defeat stochastic gradient-based defense mechanisms. It calculates the expectation of the gradient of all transformed samples and then exploits this average gradient to attack the original input image.

3.1.2 Randomization

Another line of works investigates randomization schemes. The intuition is that: IC system is robust to random perturbations. Therefore, if adversarial perturbations can be randomized into random perturbations, the robustness of the IC system would be enhanced.

3.1.2.1 Random Input Transformation & Feature Pruning

Essentially, the aforementioned stochastic gradients-based defense mechanism - random resizing [29] and random dropping neurons [94], can be separately categorized into input transformation randomization and random feature pruning as well. Moreover, the image processor proposed in [96] also resorts to input transformation randomization. Luo *et al.* [30] present another random feature pruning approach. It randomly masks feature maps computed from conventional layers. Consequently, each kernel filter extracts the feature from partial positions. It leads to learning features distributing consistently concerning the mask pattern, and the IC system can capture more information on the spatial structures of local features.

3.1.2.2 Random Noising

Liu *et al.* [97] first design Random Self-Ensemble (RSE). It inserts a noise layer before each convolutional layer in IC system. Formally, given features, the noise layer would add Gaussian noise to it. In the other perspective, this defense is equivalent to ensemble multiple noisy systems. Lecuyer *et al.* [98] establish the connection between adversarial robustness and Differential Privacy (DP) [99] together, and propose a DP based defense approach named PixelDP. PixelDP implants noising layers into IC system, which are specifically designed to enforce DP bounds of IC system. DP bound of a given system is defined to measure the change of distribution over the system’s predictions with norm-bounded changes in the input. Inspired by PixelDP, Li *et al.* [100] propose to add random noise to adversarial examples on pixel-level for diminishing adversarial perturbation. They further demonstrate that it can be robust to upper-bound of the adversarial perturbation.

3.1.2.3 Drawbacks

Essentially, the aforementioned random input transformation and random feature pruning approach either lead to shattered gradients or stochastic gradients. They can also be defeated by BPDA and EOT attacks. As for random noising, RSE provides limited formalism that is insufficient to answer the following defense design questions such as the position to add noise layers, the quantities of noise, and formal robustness boundary. Although PixelDP and Li’s [100] can provide a boundary of robustness, they are challenging to scale to large datasets due to high computational complexity. Moreover, they are mostly demonstrated for small IC systems and relatively small adversarial perturbation.

3.1.3 Denoising

Another research direction is directly to mitigate adversarial perturbations via denoising. It can be categorized into two types: pixel level denoising and feature level denoising. Pixel level denoising intends to remove adversarial perturbation on images. Feature denoising aims to alleviate the effect caused by adversarial perturbations on high level feature representation.

3.1.3.1 Pixel Level Denoising

Some researchers aim to employ a generative model to project adversarial examples into clean data manifold before sending them to classifiers. Song *et al.* [101] propose Pixel Defend. Formally, given a clean training image distribution $P(\mathbf{x}^{\text{clean}})$, the projection can be defined as follows:

$$\max P(\mathbf{x}^*) \quad \text{s.t.} \quad \|\mathbf{x} - \mathbf{x}^{\text{clean}}\|_{\infty} \leq \epsilon_{\text{defend}}, \quad (3.1)$$

where ϵ_{defend} denotes ϵ_{defend} -ball of $\mathbf{x}^{\text{clean}}$ and $P(\mathbf{x}^{\text{clean}})$ is modelled by pixelCNN [102].

Similarly, Samangouei *et al.* [31] leverage Generative Adversarial Network (GAN) [103] to alleviate the effect of adversarial perturbation and propose Defense-GAN. Defense-GAN aims to search for random vectors $\mathbf{z}$ as follows:

$$\min \quad \|f^{\text{generator}}(\mathbf{z}) - \mathbf{x}^{\text{clean}}\|_2^2, \quad (3.2)$$

where $f^{\text{generator}}$ denotes generator. Subsequently, $f^{\text{generator}}(\mathbf{z}^*)$ is sent to the classifier. Moreover, since both Pixel Defend and Defense-GAN involve generative networks, IC system becomes extremely deep, leading to the potential of gradients exploding and vanishing. Eventually, gradient information is difficult to be scraped by gradient-based attacks.

Liao *et al.* [104] further propose a High-Level Feature Guided Denoising approach named HGD. To be specific, they first design a Denoising U-Net (DUNET), denoted as $\mathcal{D} : \mathbf{x}^{\text{adv}} \rightarrow \hat{\mathbf{x}}$. It is made up of convolutional version of Denoising Auto-Encoder (DAE) [105] and U-Net [106]. Afterward, they aim to minimize representation difference extracted by l -th layer of $\mathcal{D}$ to train $\mathcal{D}$ as follows:

$$\min \quad \|\mathcal{D}_l(\mathbf{x}^{\text{clean}}) - \mathcal{D}_l(\hat{\mathbf{x}})\|, \quad (3.3)$$

where l stands for l -th layer of $\mathcal{D}$.

Xu *et al.* [91] combine denoising with adversarial example detection to enhance IC system robustness. They assume that adversarial example and the corresponding denoised sample would produce substantially different outputs computed by IC system. Formally, given a denoising approach $\mathcal{D}$, $\|f_{\theta}(x^{\text{adv}}) - f_{\theta}(\mathcal{D}(x^{\text{adv}}))\| > \|f_{\theta}(x^{\text{clean}}) - f_{\theta}(\mathcal{D}(x^{\text{clean}}))\|$.

Therefore, they mainly compare this difference with pre-defined threshold, detecting whether given sample is adversarial example or not. They employ feature squeezing approaches as $\mathcal{A}$.

Moreover, aforementioned non-differentiable pre-processing approaches in Sec. 3.1.1.1 - image quilting, TV minimization, and quantization, can be regarded as pixel level denoising as well.

3.1.3.2 Feature Level Denoising

Xie *et al.* [32] observe that adversarial perturbation results in extensive noise in the feature space even though it is tiny in the pixel space. Therefore, they propose a feature level denoising approach along with adversarial training (Sec. 3.1.4.2) to improve the robustness of IC system. Given a polluted feature map ϖ' , they try four feature denoising approaches, which are non-local means, bilateral filter, mean filter, and median filter, to generate denoised feature map ϖ .

Non-Local Means Non-local means purify ϖ' by employing a weighted average of features in all spatial locations $\mathcal{S}$ as follows:

$$\varpi_i = \frac{1}{\mathcal{K}(\varpi')} \sum_{\forall j \in \mathcal{S}} w(\varpi'_i, \varpi'_j) \cdot \varpi'_j, \quad (3.4)$$

where $\mathcal{K}$ denotes a normalization function and w represents weighting function.

They propose two forms of $\mathcal{K}(\varpi')$ and $w(\varpi'_i, \varpi'_j)$. The first form is called Gaussian sets and it is defined as follows:

$$w(\varpi'_i, \varpi'_j) = \exp \frac{1}{\sqrt{N^{\text{channel}}}} m_\gamma(\varpi'_i)^T m_\alpha(\varpi'_j), \quad (3.5)$$

$$\mathcal{K}(\varpi') = \sum_{\forall j \in \mathcal{S}} w(\varpi'_i, \varpi'_j), \quad (3.6)$$

where N^{channel} stands for the number of channels, m_γ and m_α denote mapping function parameterized by γ and α , respectively.

The second form is named dot product sets and it is formulated as follows:

$$w(\varpi'_i, \varpi'_j) = \varpi'_i{}^T \varpi'_j, \quad (3.7)$$

$$\mathcal{K}(\varpi') = N^{\text{pixel}}, \quad (3.8)$$

where N^{pixel} means the number of pixels in ϖ' . Empirical results demonstrate that dot product set can suppress noise without involving mapping function. It helps to save parameters and improve computational efficiency.

Bilateral Filter Another type of approach named bilateral filter is local version of non-local means. It is widely used for edge-preserving denoising. It is formulated as follows:

$$\varpi_i = \frac{1}{\mathcal{K}(\varpi')} \sum_{\forall j \in \Phi(i)} w(\varpi'_i, \varpi'_j) \cdot \varpi'_j, \quad (3.9)$$

where $\Phi(i)$ means a local region around pixel i . Bilateral filter can also employs the aforementioned two forms: Gaussian sets and dot product sets.

Mean Filter and Median Filter Mean filter is the simplest denoising form. It conducts average pooling of ϖ' with a stride of 1. Median filter is rarely used in deep learning but it is effective to remove salt-and-pepper noise. It is defined as follows:

$$\varpi_i = \text{median}\{\forall j \in \Phi(i) : \varpi'_j\}. \quad (3.10)$$

3.1.3.3 Drawbacks

For pixel-level denoising approaches, if the approach is smooth and differentiable, gradients computed from the IC system are often sufficient to craft adversarial examples [107]. As for Pixel Defend and Defense-GAN, their defense is mainly contributed by gradients exploding/vanishing rather than denoising. They can be solved by re-projecting denoised inputs exclusively on the clean data manifold [95]. PGD attacks under a white-box setting can defeat HGD. Moreover, prior to employing HGD is to craft enough adversarial examples and train other networks. These pre-conditions lead to higher computational complexity. For those denoising approaches utilizing adversarial detection, recent studies have demonstrated that adversarial detection methods, which rely on the assumption that the distributions of adversarial and benign examples are different, can be evaded by Carlini & Wagner’s (C&W) attack.

3.1.4 Robust Optimization

Robust optimization improves IC system reliability by modifying learning manner. Generally, this defense strategy assumes that the adversarial space is known by defenders. Afterward, they train a robust IC system f_{θ^*} that can against adversarial attacks. These approaches intend to train optimal parameters θ^* that has a minimum mean of adversarial loss as follows:

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{(\mathbf{x}^{\text{clean}}, y) \sim D} \max_{\|\mathbf{x}^{\text{adv}} - \mathbf{x}^{\text{clean}}\|_p \leq \epsilon} \mathcal{L}(\theta; \mathbf{x}^{\text{adv}}; y). \quad (3.11)$$

3.1.4.1 Lipschitz Constant Regularization

Generally, Lipschitz constant of a IC system f_{θ} is defined as follows [108]:

$$C_f = \max_{\mathbf{x}_1 \neq \mathbf{x}_2} \frac{\|H(\mathbf{x}_1) - H(\mathbf{x}_2)\|_{\infty}}{\|\mathbf{x}_1 - \mathbf{x}_2\|_2}, \quad (3.12)$$

where H denotes score function.

If H is differentiable on closed and bounded domain, then Eqn. (3.12) could be reformulated as:

$$C_f = \max_{\mathbf{x}} \|\nabla H(\mathbf{x})\|_{2, \infty}. \quad (3.13)$$

Lower bound of C_f could be approximated by evaluating dataset $\mathcal{D}$ as:

$$\max_{\mathbf{x} \in \mathcal{D}} \|\nabla H(\mathbf{x})\|_{2, \infty} \leq C_f. \quad (3.14)$$

Upper bound of C_f could be estimated directly using the product of the norm of the weight matrices [109].

$$C_f \leq \prod_{k=1}^{N^L} \|w_k\|, \quad (3.15)$$

where w_k stands for the k -th layer of f_{θ} and f_{θ} consists of N^L layers.

Weng *et al.* [33] demonstrates that C_f could be exploited to compute a minimum adversarial perturbation δ as follows:

$$\delta \geq \min_{j \neq y} \frac{H_y(\mathbf{x}) - H_j(\mathbf{x})}{2C_f}. \quad (3.16)$$

Consequently, some works are proposed to train f_θ with smaller C_f , so that minimum perturbation has to be enlarged and easily be detected [110,111]. As shown in Eqn. (3.15), upper bound of C_f is related to weight matrices. If weight matrices are sparser, C_f would be smaller, leading to a more robust IC system. Weight sparsity on the system can be achieved by involving l_1 regularization [112].

Another direction is to design a Lipschitz-constrained system by imposing a Lipschitz constraint on each layer. The Lipschitz bound of the system is the product of the bounds for each layer. Szegedy *et al.* [12] first suggest the penalization of the layer’s Lipschitz constant so that the output of each layer would not be influenced by the variation of its input significantly. Consequently, it leads to forcing the IC system to be stable. Cisse *et al.* [113] formalize this idea and propose Parseval regularization. They first provide the Lipschitz constant of standard layers such as linear layers, convolutional layers, and aggregation layers. Then, to constrain the Lipschitz constant of each layer to be smaller than one, they add orthonormality constraints for linear/convolutional layers and convex constraints in aggregation layers. Thus, the Lipschitz constant of the IC system would not grow exponentially with its depth. Tsuzuku *et al.* [114] provide tighter bounds of each component of the IC system. Afterward, they train the IC system to maximize the output margin using a modified objective function with the Lipschitz constant.

3.1.4.2 Adversarial Training

Adversarial training is first proposed in [9]. Specifically, it first generates adversarial examples $\mathbf{x}^{\text{adv}}$ crafted by FGSM. Afterward, clean samples are mixed with adversarial examples as data augmentation. Consequently, the IC system is trained with augmented data samples and expected to classify adversarial examples, improving its reliability correctly. Unfortunately, the efficiency of adversarial training is a key challenge as defenders have to craft adversarial examples first. Kurakin *et al.* [115] aims to scale adversarial training to a large dataset by following modification: i) generating adversarial examples by batch and ii) involving batch normalization [116].

Moreover, the performance of adversarial training highly depends on the attack approach exploited to craft adversarial examples and the real attack strategy applied to IC system. For instance, given an IC system adversarially-trained with FGSM adversarial

examples, it is still threatened by much more powerful attacks such as PGD and BIM, etc. Subsequently, Madry *et al.* [34] utilize PGD attack for adversarial training since PGD is appraised much more potent than single-step attack FGSM. They demonstrate that adversarial training with PGD can defend against both FGSM and PGD attacks. However, as the PGD attack is iterative rather than a single step, the time complexity is much higher than FGSM, leading to the impossibility of scaling adversarial training to large datasets such as ImageNet.

To further enhance adversarial training efficiency, Tramèr *et al.* [117] propose Ensemble Adversarial Training (EAT). They employ adversarial examples crafted by FGSM with ensemble learning for improving system robustness with lower computational complexity. To be specific, a set of IC systems $\{f_{\theta_1}, f_{\theta_2}, \dots, f_{\theta_M}\}$ with different hyperparameters is trained. Subsequently, given clean sample $\mathbf{x}^{\text{clean}}$, FGSM is employed to craft adversarial examples on these IC systems to separately generate $\{\mathbf{x}_1^{\text{adv}}, \mathbf{x}_2^{\text{adv}}, \dots, \mathbf{x}_M^{\text{adv}}\}$. Based on the transferability property of adversarial examples, $\{\mathbf{x}_1^{\text{adv}}, \mathbf{x}_2^{\text{adv}}, \dots, \mathbf{x}_M^{\text{adv}}\}$ can also fool target f_{θ} . Lastly, f_{θ} is adversarially-trained with $\{\mathbf{x}_1^{\text{adv}}, \mathbf{x}_2^{\text{adv}}, \dots, \mathbf{x}_M^{\text{adv}}\}$ to improve robustness of f_{θ} .

Another interesting approach is called free adversarial training [118] to improve the efficiency of adversarial training with PGD. This efficiency is thanks to the computation of both loss's gradient to input and system parameters just in a single backpropagation step by exploiting chain rule. As a consequence, adversarial training with PGD is highly accelerated. Zhang *et al.* [119] states that $\mathbf{x}^{\text{adv}}$ crafted by PGD attacks are only coupled with the parameters of the first layer of IC system. Subsequently, they propose You Only Propagate Once (YOPO) to reuse first layer gradient information during PGD attack, and it reduces time complexity on calculating gradient.

3.1.4.3 Drawbacks

Robust optimization-based defense approaches have the following drawbacks. Firstly, the Lipschitz constant regularization either hamper representational power or pose challenges in optimization [120]. Moreover, since IC systems are generally made up of many layers, if the bound of the Lipschitz constant of the system is computed by the product of the Lipschitz constant of each layer, the bound might be loose. Moreover, a layer with

a Lipschitz bound can only reduce the gradient norm during backpropagation. Consequently, it may lead to gradient attenuation, resulting in a much smaller Jacobian for the IC system than the one that is theoretically allowed [120, 121].

Adversarial training has the following drawbacks: i) although FGSM adversarial training is fast and it is robust in defending against FGSM attacks, it is still vulnerable to iterative or optimization-based adversarial attacks, ii) PGD adversarial training is overall the most effective countermeasure. However, it is not efficient due to the enormous computational cost required for PGD adversarial examples generation, iii) to defend against more attacks, adversarial training requires a sufficient amount and variety of adversarial examples, leading to a high time-complexity [117], iv) EAT need to train a set of IC systems to improve the robustness of target system. Moreover, EAT can be defeated by adversarial examples generated by the momentum iterative methods for multiple systems in a black-box manner [56], and v) YOPO and free adversarial training are unclear how to extend these approaches to a union of perturbations rather than PGD attack-specific.

3.1.5 Our Proposal

Although aforementioned adversarial defense approaches have demonstrated their efficacy of improving IC system robustness, most of them can be defeated by more powerful attacks or suffer from high computational complexity caused by crafting specific adversarial examples, training additional networks, and complex optimization. Little attention has been given to the direct design of robust frameworks in an *attack-agnostic manner*. That is directly enhancement the robustness of a target IC system to cope with more potential attacks. To be specific, we consider an attack-agnostic manner to satisfy the following criterion: i) coping with most well-known white-box attacks such as FGSM, PGD, C&W, Deep Fool, and black-box attacks, ii) without crafting any specific adversarial attacks, and iii) without involving non-differentiable modules for circumventing BPDA and EOT attacks.

Consequently, to enhance the intrinsic robustness of IC systems in an attack-agnostic manner, we propose a differentiable adversarial defense framework named Feature Pyramid Decoder (FPD), applicable to enhance all types of block-based IC systems. To be specific, the FPD framework implants a target IC system with both denoising and image

restoration modules to filter an input image at multiple levels. Furthermore, we analyze that network with Softmax as a classification layer’s activation function can be pretty sensitive to some perturbations. Lipschitz constant regularization would be a good choice to alleviate these perturbations. However, as explained in Section 3.1.4.1, the Lipschitz constant is generally employed to design a Lipschitz-constrained system by imposing a Lipschitz constraint on each layer. It would either reduce representational power or pose challenges in optimization. Therefore, this might lead one to wonder: is it possible to impose Lipschitz constant constraint on the final classification layer only for mitigating those drawbacks? Consequently, we deploy a Lipschitz Constant Constraint at the classification layer to limit the output variation in the face of attack perturbation. In order to train an FPD-enhanced IC system, we propose a two-phase strategy; it utilizes ϵ -neighbourhood noisy images to drive multi-task and self-supervised learning. Multi-task learning concentrates on improving both the quality of the regenerate images and the performance of final classification. Aided by the supervision of clean images, the enhanced IC system could be trained to denoise images and abstract the features from the denoised images.

Our proposal is partially related to [32], as the denoising layers in our FPD are inspired by their feature level denoising approach. Nevertheless, different from [32], the principle behind our FPD is to improve the intrinsic robustness, regardless of conducting adversarial training or not. Consequently, FPD includes denoising modules and image restoration modules and the Lipschitz constant constrained classification layer as well, establishing a multi-task and self-supervised training environment. Moreover, we employ denoising layers in a much more effective way: instead of implanting them to all blocks of the enhanced IC system, only shallow blocks are enhanced for maintaining high-level abstract semantic information.

3.2 Feature Pyramid Decoder

In this section, we introduce each component of our Feature Pyramid Decoder (FPD), shown in Figure 3.1. Firstly, we introduce the structure of the front denoising module FPD^{FD} and back denoising module FPD^{BD} . Next, the structure of the image restoration

Figure 3.1: The structure of the block-based IC system, enhanced with the proposed framework named FPD: it consists of the Lipschitz constant constrained classification layer FPD^{LCC} ; the front denoising module FPD^{FD} , the image restoration module FPD^{R} , a middle denoising layer and the back denoising module FPD^{BD} . ϵ -neighbourhood noisy samples $\mathbf{x}^{\text{noisy}}$ and original samples $\mathbf{x}^{\text{clean}}$ are used to train the FPD. Orange, blue and green blocks represent the original components of the IC system, the proposed components that implanted to the IC system, the modified components of the IC system, respectively.

module FPD^{R} is depicted. Then, we modify the classification layer of the IC system by applying the Lipschitz constant constraint (FPD^{LCC}). Finally, our two-phase training strategy is introduced, utilizing ϵ -neighbourhood noisy images with multi-task and self-supervised learning.

3.2.1 Front and Back Denoising Module

A denoising module is an IC system implanted by specific inner denoising layers. Specifically, a group of inner denoising layers is only implanted into the shallow blocks of a block-based IC system. Consequently, the shallow features are processed to alleviate noise, whereas the deep features are directly decoded, helping to keep the abstract semantic information. Meanwhile, we employ a residual connection between denoised features and original features. Most of the original features could be kept in the light of it, and it helps to amend gradient update.

(a) inner denoising layer with bottleneck (b) inner denoising layer without bottleneck (c) middle denoising layer without bottleneck

Figure 3.2: Three types of denoising layers which we have experimented on. (a) an inner denoising layer that linking two residual blocks with bottleneck, (b) an inner denoising layer that linking two residual blocks without bottleneck and (c) a middle denoising layer that denoising the input of the last part without bottleneck.

Moreover, we modify non-local means algorithm [122] by replacing the Gaussian filtering operator with a dot product operator. It could be regarded as a self-attention mechanism interpreting the relationship between pixels. Compared with the Gaussian filtering operator, the dot product operator helps improve the adversarial robustness [32]. Meanwhile, as the dot product operator does not involve extra parameters, it contributes to relatively lower computational complexity.

We explore two inner denoising structures shown in Figure 3.2a and Figure 3.2b. The corresponding performance comparison is conducted in Section 3.3.1. In our framework, the parameters of FPD^{FD} and FPD^{BD} are shared for shrinking the network size. The motivation of exploiting weight sharing mechanism has been explained in [123]: weight sharing mechanism not only reduces Memory Access Cost (MAC) but also provides more gradient updates to the reused layers from multiple parts of the network, leading to more diverse feature representations and helping FPD to generalize better.

3.2.2 Image Restoration Module

To build the restoration module, we firstly upsample feature maps from each block of FPD^{FD} (except the first block) for the image dimension consistency, and then the upsampled feature maps are fused. Finally, a group of the transposed convolutions transforms

the fused feature maps into an image with the exact resolution as the input. On the other hand, we especially find that particular noise is brought by the $\mathbf{x}'^{\text{clean}}$. To minimize its influence, another middle denoising layer is applied to the $\mathbf{x}'^{\text{clean}}$, depicted in Figure 3.2c. Contributed by the image restoration module and the denoising module, it helps establish a two-phase training strategy.

3.2.3 Lipschitz Constant Constrained Classification

As stated in our following Theorem 1, we demonstrate that the popular classification layer (Softmax σ_x is directly used as the last layer's activation function) can cause that the network could be sensitive to some perturbations. Therefore, we pose a Lipschitz constant constraint on the classification layer by adding a squeezing function with a small Lipschitz constant before the Softmax function. Currently, no network has ever adopted another output-layer activation function before σ_x in defending against adversarial examples so far.

Figure 3.3: Implementation details of Lipschitz constant constrained (FPD^{LCC}). It is implemented by involving a squeezing activation function to the output of a fully connected layer, i.e. Tanh.

Theorem 1 (the constraint on Lipschitz constant for fully-connected network)

Let NN_{FC} be a N^{class} -way- N^L -layer-fully-connected network, $\text{NN}_{\text{FC}}(\mathbf{x})_k$ be the k -th component of the network output given input $\mathbf{x}$, w_i be the weight matrix of the i -th layer of the network, and b_i be a bias matrix of the same layer. Given a noise vector δ , we can bound the variation $\mathcal{V}$ component-wisely from above by:

$$\mathcal{V}_k = |\text{NN}_{\text{FC}}(\mathbf{x})_k - \text{NN}_{\text{FC}}(\mathbf{x} + \delta)_k| \leq \frac{\exp^{\theta_k |x|} (\exp^\eta - \exp^{-\eta})}{\sum_p \exp^{\theta_p |x + \delta|}},$$

Figure 3.4: Implementation details of two-phase training strategy utilizing self-supervised and multi-task learning: the enhanced IC system FPD, in which FPD^R refers to the image restoration module; FPD^{FD} stands for the front denoising module; FPD^{BD} stands for the back denoising module; FPD^{LCC} refers to the modified classification layer; $\mathbf{x}^{\text{noisy}}$ are the samples in the ϵ -neighbourhood of each image. The first phase training is optimized by $\mathcal{L}_2(\mathbf{x}^{\text{clean}}, \mathbf{x}'^{\text{clean}})$ loss. If $\mathcal{L}_2(\mathbf{x}^{\text{clean}}, \mathbf{x}'^{\text{clean}}) > \lambda$, only the parameters of FPD^R and FPD^{FD} is updated. Once the $\mathcal{L}_2$ loss reaches the λ , the cross entropy (CE) loss $\mathcal{L}^{\text{CE}}$ with $\mathcal{L}_2$ loss jointly trains the enhanced IC system. Then, the second phase train the enhanced IC system further, jointly optimized by $\mathcal{L}^{\text{CE}}$ and $\mathcal{L}_2$ loss.

where $\theta_k|_x$ is the k -th component of the input to Softmax σ_x given input $\mathbf{x}$. Given σ_x as the activation function of the output layer, we denote the activation function of earlier layers's Lipschitz constant by C_f , and let $\eta = \max_{k=1, \dots, N^{\text{class}}} \{[w_{NL}C_f^{N^L-1} |w_{NL-1}w_{NL-2} \dots w_1\delta| + b_{NL}]_k\}$.

We postpone the proof of Theorem 1 to the Appendix A.1. The theorem clearly shows that w_{NL} and b_{NL} may have more prominent influence than $C_f^{N^L-1} |w_{NL-1}w_{NL-2} \dots w_1\delta|$ on the variation of the output $\mathcal{V}_k$, when we have $0 \leq C_f \left(\sqrt[N^L]{|w_{NL-1}w_{NL-2} \dots w_1\delta|} \right) \leq 1$ achieved by using regularization to restrict the weights getting close to zero. Therefore, we want to restrict w_{NL} and b_{NL} by utilizing a squeezing function f^{squeeze} with a small Lipschitz constant C_s before σ_x in the output layer. Consequently, this reduces η to $\max_{k=1, \dots, N^k} \{C_s C_f^{N^L-1} [|w_{NL}w_{NL-1}w_{NL-2} \dots w_1\delta|]_k\}$, potentially leading to a smaller $\mathcal{V}_k$.

Algorithm 1 Detail Training Procedures

Input:

Clean images $\mathbf{x}^{\text{clean}}$, label y , predict label $\hat{y}$, optimizer opt , updated parameters θ , seed s , other hyperparameters $\beta_1, \beta_2, \epsilon$, the enhanced IC system FPD, loss functions $\mathcal{L}_2$ and $\mathcal{L}^{\text{CE}}$, random sampler RS, threshold λ , epoch N^{epoch}

Output:

FPD

```

1: Normalize  $\mathbf{x}^{\text{clean}}$  into  $[0, 1]$ 
2: for  $i = 1$  to  $N^{\text{epoch}}$  do
3:    $\delta = \text{RS}(s)$  and update  $s$ 
4:    $\mathbf{x}^{\text{noisy}} := \mathbf{x}^{\text{clean}} + \delta$ 
5:    $\hat{y}^{(1)}, \mathbf{x}'^{(1), \text{clean}} := \text{FPD}(\mathbf{x}^{\text{noisy}})$ 
6:    $j_2 = \mathcal{L}_2(\mathbf{x}^{\text{clean}}, \mathbf{x}'^{(1), \text{clean}})$ 
7:   if  $j_2 > \lambda$  then
8:      $\text{opt}(\theta = [\text{FPD}^{\text{FD}}, \text{FPD}^{\text{R}}], \text{loss} = \beta_2 j_2)$ 
9:   else
10:     $j_1 = \mathcal{L}^{\text{CE}}(y, \hat{y}^{(1)})$ 
11:     $\text{opt}(\theta = [\text{FPD}], \text{loss} = \beta_1 j_1 + \beta_2 j_2)$ 
12:     $\hat{y}^{(2)}, \mathbf{x}'^{(2), \text{clean}} = \text{FPD}(\mathbf{x}'^{(1), \text{clean}})$ 
13:     $j_2 = \mathcal{L}_2(\mathbf{x}^{\text{clean}}, \mathbf{x}'^{(2), \text{clean}})$ 
14:     $j_1 = \mathcal{L}^{\text{CE}}(y, \hat{y}^{(2)})$ 
15:     $\text{opt}(\theta = [\text{FPD}], \text{loss} = \beta_1 j_1 + \beta_2 j_2)$ 
16:  end if
17: end for
18: return FPD

```

Therefore, the output of NN_{FC} could be more stable in the face of attack perturbation.

To thwart various attacks, we let $f^{\text{squeeze}} = \text{Tanh}(\mathbf{x})$ as our squeezing function, shown in Figure 3.3. Moreover, we empirically replace all the activation functions from ReLU to ELU; this leads to a smoother classification boundary, thus adapting to more complex distributions.

3.2.4 Training Strategy

We carefully devise our training strategy and involve uniformly sampled random noise to the clean images for further improving the enhanced IC system. Let us define the enhanced IC system FPD, in which FPD^{R} refers to the image restoration module; FPD^{FD} stands for the front denoising module; FPD^{BD} stands for the back denoising module; FPD^{LCC} refers to the modified classification layer.

To further improve the denoising and generalization capability, we suppose that the samples in the ϵ -neighbourhood of each image $\mathbf{x}^{\text{clean}}$ constitute adversarial samples candidate sets. We first randomly sample δ from uniform distribution $U(-\epsilon^{\text{noisy}}, \epsilon^{\text{noisy}})$, where $\epsilon^{\text{noisy}} \leq \epsilon$. Subsequently, we add δ to the clean images. It is impossible to use all samples in candidate sets, but the enhanced IC system will have more stable performance on classifying images in a smaller δ -neighbourhood ($[\mathbf{x}^{\text{clean}} - \delta, \mathbf{x}^{\text{clean}} + \delta]$, $0 \leq \|\delta\|_{\infty} \leq \epsilon$) after training on noisy images. The detail training procedures are described in Algorithm 1.

We propose the two-phase training to drive the self-supervised and multi-task learning for jointly optimizing the enhanced IC system. It helps the enhanced IC system learn how to denoise images and abstract features from them with a low cost and helps the enhanced IC system learn a much more accurate mapping between images and labels. As shown in Figure 3.4, the first phase mainly focuses on regenerating images, optimized by $\mathcal{L}_2(\mathbf{x}^{\text{clean}}, \mathbf{x}'^{\text{clean}})$ loss. To guarantee the quality of $\mathbf{x}'^{\text{clean}}$ used in the later training procedures, we set a threshold λ . If $\mathcal{L}_2(\mathbf{x}^{\text{clean}}, \mathbf{x}'^{\text{clean}}) > \lambda$, only the parameters of FPD^{R} and FPD^{FD} is updated for generating the higher quality $\mathbf{x}'^{\text{clean}}$. Once the $\mathcal{L}_2$ loss reaches the λ , the CE loss $\mathcal{L}^{\text{CE}}$ with $\mathcal{L}_2$ loss jointly trains the enhanced IC system. Then, the second phase focus on using the good quality $\mathbf{x}'^{\text{clean}}$ to train the enhanced IC system further, jointly optimized by $\mathcal{L}^{\text{CE}}$ loss with $\mathcal{L}_2$ loss.

3.3 Experiments

In this section, we firstly investigate the best framework structure through the exploration study. Moreover, we compare with the most related work [32] as well. In the comparison experiments, we compare the robustness between the enhanced IC system and the original one, conducting adversarial training and normal training, respectively. Since our architecture does not employ non-differentiable modules and it is without crafting any specific adversarial examples, we evaluate the efficacy of the FPD framework in defending against well-known white-box attacks FGSM, PGD, C&W, Deep Fool, and black-box attacks while assessing whether our proposal is attack-agnostic or not.

We employ MNIST, SVHN CALTECH-101 and CALTECH-256 datasets in the following experiments. MNIST consists of a training set of 60,000 samples and a testing

dataset of 10,000 samples. SVHN is a real-world colored digits image dataset. We use one of its formats, including 73,257 MNIST-like 32-by-32 images centered around a single character for training and 10,000 images for testing. For both MNIST and SVHN, we resize them to image size 64. Besides, we repeat the channel three times on MNIST for network consistency. For both CALTECH-101 and CALTECH-256, we randomly choose 866 and 1,422 images as test images resized into 224-by-224, respectively. We normalize image pixel value into $[0, 1]$. ResNet-101, ResNet-50 [3] as well as ResNeXt-50 [124] are enhanced in the following experiments. We use Pytorch to implement the whole experiment.

3.3.1 Exploration Experiments

In this section, we conduct the exploration experiments of the FPD-enhanced IC system which is based on ResNet-101 $\mathcal{F}$ on MNIST. In Table 3.1, we use l_∞ PGD attack with parameters: $\epsilon = 0.3$, step = 40, step size = 0.01 for both white-box and black-box attacks. Under the black-box condition, we separately train a simple three layers fully-connected network as the substitute network [55] for each network.

Inner Denoising Layers Implanted Positions Selection We firstly explore the position to implant the inner denoising layers. In Table 3.1, “Shallow” means that the denoising layers are implanted into the first two residual blocks. Likewise, “Deep” means that the layers are implanted into the third and fourth residual blocks. We observe that the “Shallow” outperforms ‘Deep’ on average. It may be contributed by the high-level abstract semantic information generated from the directly decoded deep features. In the following experiments, we always implant the inner denoising layers to the shallower blocks.

Denoising Approaches Selection Next, we explore the best denoising operation. In Table 3.1, no denoising layers are implanted in both the front and back denoising modules in the “Average”, “Flip”, and “Mid” denoising approaches. In these three approaches, we only focus on cleaning the $\mathbf{x}'^{\text{clean}}$ before passing to $\mathcal{F}^{\text{BD}}$. Specifically, “Average”: $\mathbf{x}'^{\text{clean}}$ and $\mathbf{x}^{\text{clean}}$ are averaged; “Flip”: $\mathbf{x}'^{\text{clean}}$ are flipped; “Mid”: the noise in $\mathbf{x}'^{\text{clean}}$

Table 3.1: Overall results of the exploration experiments with ResNet-101 on MNIST.

Inner Denoising Layer Implanted Position Selection			
Accuracy	WhiteBox	BlackBox	Average
Shallow	1.67%	31.10%	16.39%
Deep	2.08%	27.02%	14.55%
Denoising Approaches			
Average	11.04%	15.99%	13.51%
Flip	1.22%	17.34%	9.28%
Mid	0.32%	53.77%	27.05%
Mid + Inner	7.44%	42.41%	24.93%
Ablation Study			
$\mathcal{F}^{\text{FD}}$	1.67%	31.10%	16.34%
$\mathcal{F}^{\text{FD+R}}$	7.44%	42.41%	24.93%
$\mathcal{F}$	25.55%	62.72%	44.14%
Activation Functions Selection			
ReLU	0.29%	49.28%	24.79%
ELU	0.28%	61.24%	30.76%
ELU + Tanh	0.25%	69.11%	34.68%
Bottleneck Selection			
$\mathcal{F}^{2\text{IB-Mid}}$	22.21%	46.65%	34.43%
$\mathcal{F}^{2\text{I-Mid}}$	25.55%	62.72%	44.14%
No. Inner Denoising Layers Selection			
$\mathcal{F}^{2\text{IB}}$	0.04%	13.26%	6.65%
$\mathcal{F}^{4\text{IB}}$	1.97%	15.90%	8.94%
Training Strategy Selection			
$\mathcal{F}^{\text{One-Phase}}$	8.60%	51.08%	29.84%
$\mathcal{F}^{\text{Two-Phase}}$	25.55%	62.72%	44.14%
ResNet-101 Enhanced by [32]			
$\mathcal{X}$	5.72%	62.39%	32.56%

are alleviated by the middle denoising layer as depicted in Figure 3.2c. Finally, “Mid + Inner” means that we implant the two inner denoising layers to both the front and back denoising modules respectively. Meanwhile, the middle denoising layer is also utilized. Distinctly, “Mid + Inner” is all-sided robust among them to defend against both the black-box and white-box attacks, attributing to the stronger denoising capability.

Ablation Study To validate the effectiveness of $\mathcal{F}$, we perform the ablation experiments on investigating the effectiveness of each module. As shown in Table 3.1, $\mathcal{F}$ performs far better than both $\mathcal{F}^{\text{FD+R}}$ and $\mathcal{F}^{\text{FD}}$ in thwarting both white-box and black-box attacks. This overall robustness is owing to the increase of data diversity and the supervision signal brought by $\mathcal{F}$. Furthermore, $\mathcal{F}^{\text{BD}}$ can further clean the $\mathbf{x}'^{\text{clean}}$ to enhance the robustness in defending against the well-produced perturbations.

Activation Functions Selection We explore the activation functions selection. Table 3.1 indicates that ELU activation function outperforms ReLU. Furthermore, as shown in Figure 3.3, ELU with Tanh achieves better performance than ELU one with 3.92%. It demonstrates that ELU with Tanh is the suggested activation function selection.

Inner Denoising Layers Selection We also investigate the optimal number of the inner denoising layers and whether to use the bottleneck in these inner layers. In Table 3.1, $\mathcal{F}^{\text{kIB-Mid}}$: k inner denoising layers with the bottleneck as depicted in Figure 3.2a are implanted to each denoise module $\mathcal{F}^{\text{FD}}$ and $\mathcal{F}^{\text{BD}}$ respectively. Meanwhile, the middle denoising layer is used as depicted above; $\mathcal{F}^{\text{kIB}}$ is similar to $\mathcal{F}^{\text{kIB-Mid}}$ except that no middle denoising layer is involved in the framework. $\mathcal{F}^{\text{kI-Mid}}$ means that the bottleneck is not used in the inner denoising layers as depicted in Figure 3.2b. We observe that the bottleneck reduces the performance by around 10%. Moreover, although $\mathcal{F}^{\text{4IB}}$ outperforms $\mathcal{F}^{\text{2IB}}$, the enhancement is not worthy if we consider the time complexity brought by the additional denoising layers. Therefore, we use $\mathcal{F}^{\text{2I-Mid}}$ as our proposed framework in the following experiments.

Training Strategy Selection We further demonstrate the efficacy of our two-phase training strategy $\mathcal{F}^{\text{Two-Phase}}$ as depicts in Figure 3.4. We mainly compare $\mathcal{F}^{\text{Two-Phase}}$ with one-phase training strategy $\mathcal{F}^{\text{One-Phase}}$, i.e., the first training phase (describes in Section 3.2.4). Results show that $\mathcal{F}^{\text{Two-Phase}}$ could achieve higher performance than $\mathcal{F}^{\text{One-Phase}}$ with 14.3%.

Comparison with the Related Work The denoising approach proposed in [32] is similar to our denoising layers in FPD. Therefore, we conduct a comparison experiment with [32] as well. In Table 3.1, $\mathcal{X}$ represents the enhanced IC by [32]. We observe that our $\mathcal{F}^{\text{2I-Mid}}$ outperforms $\mathcal{X}$. Especially, the performance of thwarting the white-box attack is about 20% higher.

Table 3.2: Robustness evaluation results (Accuracy %, Attack Time (min)) in thwarting the white-box attacks with ResNet-101 on MNIST.

Network Name	$l_\infty(\epsilon = 0.3)$					
	FGSM		PGD		C&W	
	Acc	T(m)	Acc	T(m)	Acc	T(m)
$\mathcal{O}$	4%	0.85	0%	46.42	0%	56.75
$\mathcal{O}^{\text{FGSM}}$	43%	0.95	0%	66.43	36.63%	39.57
$\mathcal{O}^{\text{PGD}}$	92%	1.85	76%	68.37	89.88%	42.92
$\mathcal{F}$	31%	1.5	0%	72.3	88%	212
$\mathcal{F}^{\text{FGSM}}$	42%	0.95	0.87%	119	46.12%	181.6
$\mathcal{F}^{\text{PGD}}$	87%	1.6	64.03%	115.85	78%	160

	$l_2(\epsilon = 1.5)$				l_2					
Network Name	FGSM		PGD		C&W		DeepFool		Average	
	Acc	T(m)	Acc	T(m)	Acc	T(m)	Acc	T(m)	Acc	T(m)
$\mathcal{O}$	94%	0.95	82%	53.25	15%	1183.67	6%	364.13	27.57%	243.72
$\mathcal{O}^{\text{FGSM}}$	100%	0.95	80%	64.22	74%	1177.27	6%	365.17	48.52%	244.94
$\mathcal{O}^{\text{PGD}}$	98%	1.83	92%	68.32	93%	1193.5	9%	344.38	78.56%	245.88
$\mathcal{F}$	98%	1.73	95%	98.58	95%	1134.38	9.62%	911.9	59.51%	347.48
$\mathcal{F}^{\text{FGSM}}$	97%	2.25	95%	113.97	97%	1047.83	33.16%	907.3	58.74%	338.99
$\mathcal{F}^{\text{PGD}}$	100%	3.17	97%	108.5	100%	1043.25	11.87%	912.37	76.84%	334.96

3.3.2 Comparison Experiments

A series of comparison experiments² is implemented to further evaluate FPD-enhanced IC system performance on MNIST, SVHN, CALTECH-101, and CALTECH-256.

Notation and Implementation Details Firstly, the following notations are defined for accurate description: $\mathcal{F}$ represents the enhanced IC system; $\mathcal{O}$ is the original IC system; $\mathcal{F}^{\text{PGD}}$ and $\mathcal{O}^{\text{PGD}}$ is adversarial trained by l_∞ PGD (on MNIST: $\epsilon=0.3$, step=100 and step length=0.01; on SVHN: $\epsilon=8/256.0$, step=40 and step length=2/256.0); $\mathcal{F}^{\text{FGSM}}$ and $\mathcal{O}^{\text{FGSM}}$ is adversarial trained by l_∞ FGSM (on MNIST: $\epsilon=0.3$). All results are achieved with the batch size 100, running on the RTX Titan.

On MNIST For sufficient evaluation, FPD applied to ResNet-101 on MNIST is first evaluated. Two performance metrics are mainly employed: classification accuracy and attack time. Longer attack time can be a result of a monetary limit. In this perspective, longer attacking time may result in excess of time and monetary limit. The attacker may

²Adversarial-robustness-toolbox [125], a tool for testing the network’s robustness of defending against various attacks, is employed

Table 3.3: l_2 Metrics: robustness evaluation results (Accuracy(%), Attack Time(min)) in thwarting the black-box attacks with ResNet-101 on MNIST. $\mathcal{F}$ -Average ($\mathcal{O}$ -Average) measure the average performance of $\mathcal{F}$, $\mathcal{F}^{\text{FGSM}}$, $\mathcal{F}^{\text{PGD}}$ ($\mathcal{O}$, $\mathcal{O}^{\text{FGSM}}$, $\mathcal{O}^{\text{PGD}}$).

Substitute	Attack	$\mathcal{F}$	$\mathcal{F}^{\text{FGSM}}$	$\mathcal{F}^{\text{PGD}}$	$\mathcal{F}$ -Average
$\mathcal{O}$	FGSM	(99%,0.5)	(99%,1.47)	(100%,2.1)	(99.33%,1.36)
	PGD	(99%,62.23)	(99%,57.82)	(100%,66.9)	(99.33%,62.32)
$\mathcal{F}$	FGSM	$\times$	(98%,2.82)	(100%,1.9)	(99%,2.36)
	PGD	$\times$	(98%,55.23)	(100%,108.7)	(99%,81.97)
$\mathcal{O}^{\text{FGSM}}$	FGSM	(99%,1.8)	(99%,1.32)	(100%,1.18)	(99.33%,1.43)
	PGD	(99%,42.95)	(99%,71.25)	(100%,69.22)	(99.33%,61.14)
$\mathcal{F}^{\text{FGSM}}$	FGSM	$\times$	$\times$	$\times$	$\times$
	PGD	$\times$	$\times$	$\times$	$\times$
$\mathcal{O}^{\text{PGD}}$	FGSM	(99%,1.1)	(99%,1.67)	(100%,1.62)	(99.33%,1.46)
	PGD	(98%,47.75)	(98%,80.9)	(100%,59.1)	(98.67%,62.58)
$\mathcal{F}^{\text{PGD}}$	FGSM	$\times$	$\times$	$\times$	$\times$
	PGD	$\times$	$\times$	$\times$	$\times$
Average		(98.83%,26.06)	(98.63%,34.06)	(100%,38.84)	(99.17%,34.33)

Substitute	Attack	$\mathcal{O}$	$\mathcal{O}^{\text{FGSM}}$	$\mathcal{O}^{\text{PGD}}$	$\mathcal{O}$ -Average
$\mathcal{O}$	FGSM	$\times$	(100%,0.32)	(99%,0.32)	(99.5%,0.32)
	PGD	$\times$	(100%,42.87)	(98%,42.57)	(99%,42.72)
$\mathcal{F}$	FGSM	(100%,2.05)	(100%,2.8)	(99%,4.8)	(99.67%,3.22)
	PGD	(100%,167.03)	(100%,127.58)	(99%,79.12)	(99.67%,124.57)
$\mathcal{O}^{\text{FGSM}}$	FGSM	$\times$	$\times$	$\times$	$\times$
	PGD	$\times$	$\times$	$\times$	$\times$
$\mathcal{F}^{\text{FGSM}}$	FGSM	(100%,2.38)	(100%,2.39)	(98%,2.52)	(99.33%,2.43)
	PGD	(100%,117.68)	(100%,158.17)	(98%,159.27)	(99.33%,145.04)
$\mathcal{O}^{\text{PGD}}$	FGSM	$\times$	$\times$	$\times$	$\times$
	PGD	$\times$	$\times$	$\times$	$\times$
$\mathcal{F}^{\text{PGD}}$	FGSM	(100%,2.67)	(100%,2.42)	(98%,3.18)	(99.33%,2.76)
	PGD	(100%,158.07)	(100%,116.53)	(98%,86.93)	(99.33%,120.51)
Average		(100%,68.02)	(100%,56.64)	(98.38%,47.34)	(99.40%,55.20)

surrender the attack. Therefore, attackers would spend more time attacking networks, which may protect the networks from another perspective.

Various white-box attacks are employed to attack $\mathcal{F}$, $\mathcal{O}$, $\mathcal{F}^{\text{PGD}}$, $\mathcal{O}^{\text{PGD}}$, $\mathcal{F}^{\text{FGSM}}$ and $\mathcal{O}^{\text{FGSM}}$. Following attacks are considered, including l_2 PGD, l_2 FGSM, l_∞ PGD and l_∞ FGSM. $\epsilon=1.5$ and 0.3 are set to bound the permutations for l_2 and l_∞ norm. Both l_2 PGD and l_∞ PGD are set to attack for 100 iterations and each step length is 0.1.

As shown in Table 3.2, generally, $\mathcal{F}$ and its adversarial trained $\mathcal{F}^{\text{FGSM}}$ outperform $\mathcal{O}$ and $\mathcal{O}^{\text{FGSM}}$ in accuracy around 32% and 10%, respectively. $\mathcal{O}^{\text{PGD}}$ seems slightly more robust than $\mathcal{F}^{\text{PGD}}$. However, as revealed by the average attack time, more computational time (around 89 min) is spent on attacking $\mathcal{F}^{\text{PGD}}$. In particular, the overall time spent on attacking $\mathcal{F}$, its adversarial trained networks $\mathcal{F}^{\text{FGSM}}$, $\mathcal{F}^{\text{PGD}}$ are longer than $\mathcal{O}$, $\mathcal{O}^{\text{FGSM}}$

Table 3.4: l_∞ Metrics: robustness evaluation results (Accuracy(%), Attack Time(min)) in thwarting the black-box attacks with ResNet-101 on MNIST. $\mathcal{F}$ -Average ($\mathcal{O}$ -Average) measure the average performance of $\mathcal{F}$, $\mathcal{F}^{\text{FGSM}}$, $\mathcal{F}^{\text{PGD}}$ ($\mathcal{O}$, $\mathcal{O}^{\text{FGSM}}$, $\mathcal{O}^{\text{PGD}}$).

Substitute	Attack	$\mathcal{F}$	$\mathcal{F}^{\text{FGSM}}$	$\mathcal{F}^{\text{PGD}}$	$\mathcal{F}$ -Average
$\mathcal{O}$	FGSM	(50%,0.4)	(87%,1.77)	(100%,2.08)	(79%,1.42)
	PGD	(65%,44.23)	(87%,50.42)	(99%,75.95)	(83.67%,56.87)
$\mathcal{F}$	FGSM	$\times$	(80%,2.95)	(94%,1.95)	(87%,2.45)
	PGD	$\times$	(84.7%,128.25)	(98.96%,113.78)	(91.83%,121.02)
$\mathcal{O}^{\text{FGSM}}$	FGSM	(57%,1.82)	(90%,1.23)	(100%,1.23)	(82.33%,1.43)
	PGD	(72%,44.2)	(87%,87.38)	(100%,76.73)	(86.33%,69.44)
$\mathcal{F}^{\text{FGSM}}$	FGSM	$\times$	$\times$	$\times$	$\times$
	PGD	$\times$	$\times$	$\times$	$\times$
$\mathcal{O}^{\text{PGD}}$	FGSM	(39%,1.1)	(54%,1.7)	(90%,0.87)	(61%,1.22)
	PGD	(33%,48.49)	(58%,82.98)	(87%,66.72)	(59.33%,66.06)
$\mathcal{F}^{\text{PGD}}$	FGSM	$\times$	$\times$	$\times$	$\times$
	PGD	$\times$	$\times$	$\times$	$\times$
Average		(52.67%,23.37)	(78.46%,44.59)	(96.12%,42.41)	(75.75%,36.79)

Substitute	Attack	$\mathcal{O}$	$\mathcal{O}^{\text{FGSM}}$	$\mathcal{O}^{\text{PGD}}$	$\mathcal{O}$ -Average
$\mathcal{O}$	FGSM	$\times$	(51%,0.58)	(98%,0.58)	(74.5%,0.58)
	PGD	$\times$	(67%,18.13)	(99%,13.13)	(83%,15.63)
$\mathcal{F}$	FGSM	(47%,1.3)	(58%,2.82)	(95%,4.82)	(66.67%,2.98)
	PGD	(69.07%,167.9)	(77.05%,90.92)	(99%,80.5)	(81.71%,113.11)
$\mathcal{O}^{\text{FGSM}}$	FGSM	$\times$	$\times$	$\times$	$\times$
	PGD	$\times$	$\times$	$\times$	$\times$
$\mathcal{F}^{\text{FGSM}}$	FGSM	(42%,2.37)	(58%,2.47)	(96%,2.63)	(65.33%,2.49)
	PGD	(59.66%,122.22)	(61.78%,163.13)	(98%,166.03)	(73.15%,150.46)
$\mathcal{O}^{\text{PGD}}$	FGSM	$\times$	$\times$	$\times$	$\times$
	PGD	$\times$	$\times$	$\times$	$\times$
$\mathcal{F}^{\text{PGD}}$	FGSM	(53%,4.22)	(79%,4.27)	(94%,4.23)	(75.33%,4.24)
	PGD	(55.34%,168.58)	(93.84%,167.87)	(90.99%,167.7)	(80.06%,168.05)
Average		(54.35%,77.77)	(68.21%,56.27)	(96.25%,54.95)	(72.94%,62.99)

and $\mathcal{O}^{\text{PGD}}$ around 104 min, 94 min and 89 min. The above results have demonstrated that $\mathcal{F}$ and its adversarial trained networks are harder to be attacked.

Afterward, FPD applied to ResNet-101 attacked by black-box attacks is also evaluated on MNIST. As stated in [126], most of the setting of the black-box attack is that the attacker calls the target system with limited queries. Limited queries can result from attacking time limit and monetary limit if the attacker incurs a cost for each query. Therefore, as mentioned above, the author believes that a longer attacking time (more queries) may result in excess time and monetary limit. The attacker may surrender the attack. Therefore, attackers spend more time attacking networks, which may protect the networks from another perspective.

In black-box attacks, instead of relying on gradients of the network, a corresponding

Table 3.5: l_∞ Metrics: Robustness evaluation results (Accuracy %) in thwarting the black-box attacks with ResNet-101, ResNet-50 and ResNeXt-50 on SVHN.

Network		Clean	ResNet-101				
			Substitute: $\mathcal{O}$			Substitute: $\mathcal{F}$	
			FGSM	PGD	C&W	FGSM	PGD
		Acc	Acc	Acc	Acc	Acc	Acc
ResNet-101	$\mathcal{O}^{\text{PGD}}$	89%	87%	87%	86%	80%	86%
	$\mathcal{F}^{\text{PGD}}$	84%	82%	83%	84%	64%	78%
ResNet-50	$\mathcal{O}^{\text{PGD}}$	85%	81%	83%	88%	78%	81%
	$\mathcal{F}^{\text{PGD}}$	89%	87%	88%	89%	77%	87%
ResNeXt-50	$\mathcal{O}^{\text{PGD}}$	96%	92%	93.48%	96%	86%	93.45%
	$\mathcal{F}^{\text{PGD}}$	86%	80%	84%	86%	74%	82%
Average		88.17%	84.83%	86.41%	88.17%	76.5%	84.58%

Network		Clean	ResNet-50				
			Substitute: $\mathcal{O}$			Substitute: $\mathcal{F}$	
			FGSM	PGD	C&W	FGSM	PGD
		Acc	Acc	Acc	Acc	Acc	Acc
ResNet-101	$\mathcal{O}^{\text{PGD}}$	89.00%	86%	87%	86%	81%	88.01%
	$\mathcal{F}^{\text{PGD}}$	84.00%	82%	83%	84%	76%	83%
ResNet-50	$\mathcal{O}^{\text{PGD}}$	85.00%	80%	82%	88%	76%	80%
	$\mathcal{F}^{\text{PGD}}$	89.00%	87%	88%	89%	62%	71%
ResNeXt-50	$\mathcal{O}^{\text{PGD}}$	96.00%	92%	94.97%	96%	90%	92%
	$\mathcal{F}^{\text{PGD}}$	86.00%	84%	84.81%	86%	80%	84%
Average		88.17%	85.17%	86.63%	88.17%	77.50%	83%

Network		Clean	ResNeXt-50					
			Substitute: $\mathcal{O}$			Substitute: $\mathcal{F}$		Average
			FGSM	PGD	C&W	FGSM	PGD	
		Acc	Acc	Acc	Acc	Acc	Acc	Acc
ResNet-101	$\mathcal{O}^{\text{PGD}}$	89.00%	84.00%	84.00%	86.00%	92.00%	85.12%	85.68%
	$\mathcal{F}^{\text{PGD}}$	84.00%	80.00%	82.00%	84.00%	82.00%	82.00%	80.60%
ResNet-50	$\mathcal{O}^{\text{PGD}}$	85.00%	92.00%	92.00%	88.00%	92.00%	92.00%	84.87%
	$\mathcal{F}^{\text{PGD}}$	89.00%	86.00%	90.00%	88.00%	92.00%	92.00%	84.87%
ResNeXt-50	$\mathcal{O}^{\text{PGD}}$	96.00%	84.00%	86.00%	92.00%	92.00%	94.00%	91.59%
	$\mathcal{F}^{\text{PGD}}$	86.00%	66.00%	62.00%	86.00%	80.00%	82.00%	80.05%
Average		88.17%	82.00%	82.67%	87.33%	88.33%	87.85%	84.61%

substitute of the network is directly attacked to generate the adversarial samples. For each $\mathcal{F}$ -based network (i.e. $\mathcal{F}$, $\mathcal{F}^{\text{FGSM}}$ and $\mathcal{F}^{\text{PGD}}$), four networks $\mathcal{O}$, $\mathcal{O}^{\text{FGSM}}$, $\mathcal{O}^{\text{PGD}}$, $\mathcal{F}$ are selected as the substitute. Similarly, $\mathcal{F}$, $\mathcal{F}^{\text{FGSM}}$, $\mathcal{F}^{\text{PGD}}$, $\mathcal{O}$ are used as the substitute of each $\mathcal{O}$ -based network (i.e. $\mathcal{O}$, $\mathcal{O}^{\text{FGSM}}$ and $\mathcal{O}^{\text{PGD}}$). Afterwards, each substitute with two different norms l_2 and l_∞ is attacked. For each norm, two attack approaches are utilized: FGSM and PGD. $\epsilon=1.5$ and 0.3 are set to bound the perturbations for l_2 and l_∞ norm. Both l_2 PGD and l_∞ PGD are set to attack for 100 iterations and each step length is 0.1.

Overall results of all six networks that attacked by black-box attack with l_2 and l_∞ norm are reported in Table 3.3 and Table 3.4, respectively. For l_2 norm, compared with

Table 3.6: Robustness evaluation results (Accuracy %) in thwarting the white-box attacks with ResNet-101, ResNet-50 and ResNeXt-50 on SVHN.

Network Name		WhiteBox					
		$L_\infty(\epsilon = 8/256.0)$			L_2		
		FGSM	PGD	C&W	C&W	DeepFool	Average
		Acc	Acc	Acc	Acc	Acc	Acc
ResNet-101	$\mathcal{O}$	1%	0%	0%	0%	28%	5.8%
	$\mathcal{O}^{\text{PGD}}$	57%	36%	39%	1%	3%	27.2%
	$\mathcal{F}$	44%	44%	71%	62.22%	53.25%	54.89%
	$\mathcal{F}^{\text{PGD}}$	48%	47%	72.57%	77.7%	57%	60.45%
ResNet-50	$\mathcal{O}$	4%	0%	0%	0%	34%	7.6%
	$\mathcal{O}^{\text{PGD}}$	55%	26%	28%	0%	11%	24%
	$\mathcal{F}$	33%	30%	61%	52.03%	36.78%	42.56%
	$\mathcal{F}^{\text{PGD}}$	39%	35%	70%	73.3%	45.38%	52.54%
ResNeXt-50	$\mathcal{O}$	13%	0%	0%	0.4%	51.24%	12.93%
	$\mathcal{O}^{\text{PGD}}$	58%	36%	46%	4.5%	24.27%	33.75%
	$\mathcal{F}$	80%	80%	86%	86%	83.17%	83.03%
	$\mathcal{F}^{\text{PGD}}$	80%	78%	86%	84.15%	84%	82.43%

$\mathcal{O}$ -based networks on average, our $\mathcal{F}$ -based networks can achieve comparable results. As for l_∞ norm, the average accuracy of $\mathcal{F}$ -based networks is higher than $\mathcal{O}$ -based networks as well. Particularly, $\mathcal{F}^{\text{FGSM}}$ is more robust than $\mathcal{O}^{\text{FGSM}}$ around 10%. Moreover, it is noticeable that $\mathcal{F}$ -based networks are substituted by $\mathcal{O}$ -based network and vice-versa. The average results reveal that the computational time of attacking $\mathcal{F}$ -based substituted network is longer than $\mathcal{O}$ -based substituted network around 20 min (l_2) and 26 min (l_∞). Taking into account the time complexity, the result has demonstrated that attackers have to spend more time generating adversarial examples for $\mathcal{F}$ -based networks.

$\mathcal{F}$ -based networks generally require longer attacking time. The reason is that $\mathcal{F}$ -based networks are much more robust. This robustness stems from the efficacy of our denoising modules, Lipschitz constant constraint classification layer, and two-phase training strategy. Denoising modules can process shallow features to alleviate noise, whereas directly decode deep features to keep the abstract semantic information. In the light of Lipschitz constant constraint classification layer, the output of $\mathcal{F}$ -based networks could be more stable in the face of attack perturbation. As for training strategy, aided by the supervision of clean images, $\mathcal{F}$ -based networks could be trained to denoise images and abstract the features from the denoised images. Therefore, attackers have to spend a long time searching for adversarial examples that can circumvent those defenses.

Figure 3.5: Adversarial images (a) vs. the output of image restoration module from adversarial images (b) and clean images (c). Images are reproduced from the data in Table 3.6 (enhanced ResNet-101 attacked by PGD).

On SVHN The ability of FPD to enhance various block-based IC systems on colored samples: ResNet-101, ResNet-50, ResNeXt-50 are also investigated. A series of white-box and black-box attacks is utilized to attack $\mathcal{F}$, $\mathcal{O}$, $\mathcal{F}^{\text{PGD}}$ and $\mathcal{O}^{\text{PGD}}$ for each block-based IC systems. Initially, FPD performance in thwarting black-box attacks is evaluated. As shown in Table 3.5, $\mathcal{O}$ and $\mathcal{F}$ of each block-based IC systems are employed as substitutes. l_∞ FGSM and l_∞ PGD are adopted to attack them. Besides, it also shows that $\mathcal{O}$ is hard to defend against a l_∞ C&W attack, depicted in Table 3.6. Therefore, l_∞ C&W is additionally adopted to attack substitute $\mathcal{O}$, to further evaluate FPD. As for white-box attacks, following attacks are employed: l_∞ FGSM, l_∞ PGD, l_∞ C&W, l_2 DeepFool and l_2 C&W. $\epsilon=8/256.0$ is set for above-mentioned attacks and PGD is set to attack for 40 iterations with step length $2/256.0$.

The following remarks on results as shown in Table 3.5 and Table 3.6 are concluded. Firstly, in defending against white-box attacks, $\mathcal{F}$ and the adversarial trained $\mathcal{F}^{\text{PGD}}$ far outperform $\mathcal{O}$ and $\mathcal{O}^{\text{PGD}}$ in accuracy for all the block-based IC systems, especially in ResNet-101 and ResNeXt-50. It shows that the performance of $\mathcal{F}^{\text{PGD}}$ is not exactly satisfactory under black-box attacks, yet the outcome is not very surprising. As shown in Table 3.6, $\mathcal{F}$ -based networks achieve a high accuracy under white-box attacks. Therefore, when these attacks are applied to $\mathcal{F}$ substitute, some attacks effectively fail, returning a large number of clean samples as adversarial examples. Given that $\mathcal{F}^{\text{PGD}}$ has a lower

accuracy than $\mathcal{O}^{\text{PGD}}$ on clean samples for ResNet-101 and ResNeXt-50 (as depicted in Table 3.5), $\mathcal{F}$ -based networks achieve this biased performance under black-box attacks.

The output of image restoration module is shown in Figure 3.5. Adversarial images are well “denoised” by comparing Figure 3.5a with 3.5b. Figure 3.5b and 3.5c illustrate that the module output generated by adversarial and clean images are quite similar. It guarantees that restoration module could generate similar images from both adversarial and clean images for FPD^{BD} , leading to more robust performance in defending against attacks.

On CALTECH-101 & CALTECH-256 The efficacy of FPD on ResNet-101 is further demonstrated on high dimensional dataset CALTECH-101 and CALTECH-256, attacked by l_∞ PGD attack for 40 iterations. For this attack, ϵ is set to $8/256.0$ and step length to $2/256.0$. To be specific, on CALTECH-101, $\mathcal{F}$ achieves 61.78% under PGD attack. It outperforms $\mathcal{O}$ around 34.64%. On CALTECH-256, our ResNet-101 $\mathcal{F}$ achieve 49.79% accuracy against 0.00% of the original one $\mathcal{O}$.

Summary In summary, the above results have demonstrated that the FPD-enhanced IC system is much more robust than non-enhanced versions on MNIST and high dimensional datasets CALTECH-101 and CALTECH-256. Moreover, on MNIST, we observe that attacking FPD-enhanced IC systems requires a much longer attacking time. It hints that FPD-enhanced IC systems are much more robust. On colored dataset SVHN, the performance under black-box attacks is not exactly satisfactory. However, considering the performance in thwarting white-box attacks, the FPD-enhanced IC system performs far better than non-enhanced versions.

3.4 Chapter Summary

In this chapter, we have given attention to the direct design of robust frameworks. We have presented a novel attack-agnostic adversarial defense framework named Feature Pyramid Decoder (FPD) to enhance the intrinsic robustness of the block-based IC system. Besides, we have devised a novel two-phase training strategy for strategically training the enhanced IC system. We have investigated the best structure of our FPD and confirmed

the settings of the training strategy through the exploration experiments. We have performed extensive experiments under well-known white-box attacks and black-box attacks on MNIST, SVHN, and CALTECH datasets. Moreover, results have demonstrated that the proposed FPD framework can enhance the intrinsic robustness of the implanted IC system in an attack-agnostic manner. Moreover, if adversarial training is performed, the FPD-enhanced system has demonstrated a better performance than their non-enhanced version.

Chapter 4

Knowledge Transfer via Meta-Learning against Data Scarcity¹

As explained in Sec. 2.2, the Radio-Frequency Based Human Activity Recognition (RF-HAR) system is regarded as the most promising solution for performing Human Activity Recognition (HAR), giving credit to properties of RF sensing like privacy protection and energy-efficiency. Unfortunately, RF sensing is sensitive to environmental changes in practice, and RF datasets need online labeling during collection. Consequently, existing solutions to the RF-HAR system suffer from a laborious data collection process for adapting to new environments. In this chapter, we employ knowledge transfer for relieving the labeling stress for RF-HAR systems being adaptive to new environments. Moreover, we propose RF-Net, a unified meta-learning based knowledge transfer framework for Radio-Frequency (RF) enabled one-shot Human Activity Recognition (HAR), delivering the capability of being adaptive to new environments with very few labeled data. We first provide an overview of related work. Afterward, we describe our meta-learning framework, intending to endow the capability of fast environment adaptation. Subsequently, we introduce the dual-path HAR (base) network to learn influential features from general RF signals. Finally, we validate the efficacy of RF-Net by conducting extensive experiments on representative RF sensing techniques and in many indoor environments.

¹The work in this chapter has been published in [127]

4.1 Related Work

To improve Radio-Frequency Based Human Activity Recognition (RF-HAR) system performance in the face of environment adaptation and data scarcity, environment invariant systems [26, 86, 128] and knowledge transfer [25, 35, 36] are the main research directions. Environment invariant systems hope to retain HAR accuracy by handcrafting environment invariant features without re-training in a new environment. However, these features depend on prior information, such as the position and orientation of a target, likely incurring another laborious process in obtaining this information. Therefore, we mainly study knowledge transfer based approaches for achieving environment adaptation under data scarcity. Related studies [25, 35, 36] aim to mimic human behavior to learning knowledge from known tasks and adapt it to new tasks even with limited supervised information. Since they are either based on transfer learning or meta-learning, we mainly describe transfer learning and meta-learning in the following sections.

We recap notations used in knowledge transfer. Formally, a domain $\mathcal{U} = \{\mathcal{X}, P(X)\}$ is made up of a feature space $\mathcal{X}$ and a marginal distribution $P(X)$, where $X = \{\mathbf{x} | \mathbf{x}_i \in \mathcal{X}, i = 1, \dots, N\}$. Given a specific domain $\mathcal{U}$, a task $\mathcal{T} = \{\mathcal{Y}, f\}$ involves a label space $\mathcal{Y}$ and a system f learned from data $P(X)$, where $f(\mathbf{x})$ is equivalent to $P(y|\mathbf{x})$. In RF-HAR environment adaptation under data scarcity, we consider the following assumptions: i) marginal distribution of each environment differs, ii) substantial labeled source-task data samples are provided, iii) limited labeled target-task data samples are known, iv) label space is the same across all the tasks, and v) information of multiple source environments are provided.

4.1.1 Transfer Learning

To perform knowledge transfer, transfer learning can be summarized into following interpretations: data-based [129, 130], feature-based [131–133], model-based [134, 135], and adversarial learning-based, etc. Since related studies [25, 35] employ model-based and adversarial learning-based transfer learning, we mainly detail these two directions in the following sections.

4.1.1.1 Model-Based

Model-based transfer learning is also named parameter-based transfer learning. It assumes that certain model-level knowledge, such as architecture, parameters, prior information, can be shared across tasks. Knowledge of a system f^{source} well trained in source task, can be transferred to target task, helping train f^{target} with limited labeled target-task data samples.

Evgeniou *et al.* [136] decompose system parameters θ into task-specific and task-agnostic as follows:

$$\theta^{\text{source}} = \theta^{\text{invariant}} + \eta^{\text{source}}, \quad (4.1)$$

$$\theta^{\text{target}} = \theta^{\text{invariant}} + \eta^{\text{target}}, \quad (4.2)$$

where $\theta^{\text{invariant}}$ denotes task-agnostic system parameters, η^{source} and η^{target} separately represent task-specific system parameters in source and target tasks. They intend to extract $\theta^{\text{invariant}}$ from f^{source} and then employ it to f^{target} .

For neural networks based system, system parameter fine-tuning is a very effective approach. Intuitively, the system's parameters trained in the source task could be directly shared with the system in the target task. Yosinski *et al.* [137] and Mou *et al.* [138] evaluate transferability of each layer in Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), respectively. They all state that deeper layers are not suitable directly transferred to the target system. Therefore, parameter sharing is generally conducted as follows: shallow layers of f^{source} are shared to f^{target} , and then the deepest few layers of f^{target} are fine-tuned. Zhang *et al.* [35] employ this parameter sharing scheme to perform RF-HAR across different environments (deployment sites). To be specific, they assume that the same network architecture is used across tasks. After training f^{source} , they directly copy weights of f^{source} to initialize f^{target} , and then fine-tune f^{target} with fewer labeled source-task data samples.

4.1.1.2 Adversarial Learning-Based

Adversarial learning-based transfer learning is inspired by Generative Adversarial Network (GAN) [103]. They assume that an ideal feature representation shall not involve

any discriminative domain information, and they intend to produce the domain-invariant feature representation. Before diving into adversarial learning-based transfer learning, let us review the original GAN. Formally, GAN is inspired by a two-player game, consisting of two networks, a generator $f^{\text{generator}}$ and a discriminator $f^{\text{discriminator}}$. The objective of GAN is from two sides: i) $f^{\text{generator}}$ generates a replica of true data to cheat $f^{\text{discriminator}}$ and ii) $f^{\text{discriminator}}$ intends to distinguish true data and fake data. Essentially, GAN is a two-player min-max game.

Motivated by original GAN, adversarial learning-based transfer learning generally employ a feature extractor $f^{\text{extractor}}$, a label predictor $f^{\text{predictor}}$, and a domain classifier f^{domain} [139–142]. Specifically, $f^{\text{extractor}}$ plays role like generator for producing domain-invariant feature representation. $f^{\text{predictor}}$ aims to predict with extracted domain-invariant feature representations. f^{domain} works like a discriminator to distinguish whether the extracted features learned from the source domain or target domain. Consequently, $f^{\text{extractor}}$ and $f^{\text{predictor}}$ can be deployed to target domain. Jiang *et al.* [25] propose an adversarial RF-HAR network named EI for performing environment adaptation. It consists of a feature extractor, an activity recognizer, and a domain classifier. EI intends to eliminate environment-related information and extracts environment invariant feature representation.

4.1.1.3 Drawbacks

[25, 35] have one major drawback. Since they all focus on knowledge transfer to a single target environment, the whole fine-tuning process must be performed again with a substantial amount of labeled data when applied to a new target environment. Therefore, extra efforts in collecting data are still required. [26] demonstrates that the performance of [25, 35] is degraded severely in a new target environment if provided labeled data samples in the target environment are insufficient. To better tackle this challenge, we need to consider recently emerged alternatives and meta-learning provides an alternative.

4.1.2 Meta-Learning

In conventional supervised machine learning, given specific domain $\mathcal{U}$, single task $\mathcal{T}$ can provide training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)_{i=1}^N\}$ and a loss function $\mathcal{L}$, then a predictive system

f_θ parameterized by θ is trained by solving the following objective function:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\mathcal{D}; \theta; \xi^{\text{train}}), \quad (4.3)$$

where $\mathcal{L}$ is loss function to measure the difference between ground-truth and predictive result, ξ^{train} is training factors upon *how to learn* θ , such as optimizer selection and weight initialization, etc.

Conventional machine learning assumes that Eqn. (4.3) has to be solved for each single task with corresponding $\mathcal{D}$ and pre-specific ξ^{train} . Unfortunately, ξ^{train} can have a drastic impact on generalization like data efficiency and the high computational complexity, and so on. Moreover, human-level intelligence requires that a learning system mimics human behavior to learn from known tasks and adapt to new tasks quickly with only a few labeled samples.

Meta-learning provides a paradigm where the learning system achieves experience over the distribution of related tasks – and uses this experience to improve its learning performance on unseen tasks. It learns ξ^{meta} named *meta knowledge*, rather than assuming it is fix and pre-specific. In our context, the key idea of meta-learning can be summarized into twofold: i) *meta-training*: learning knowledge from performing RF-HAR in source environments with rather abundant data, and ii) *meta-testing*: exploiting accumulated knowledge in all unseen target environments, requiring only a minimum level of target labeled data samples.

Essentially, given a task $\mathcal{T}$ sampled from $P(\mathcal{T})$, task is defined as $\mathcal{T} = \{\mathcal{D}, \mathcal{L}\}$. Note that each task is generally denoted as N^{class} -way N^{shot} -shot classification task if it involves N^{shot} labeled samples per class and the number of different classes is N^{class} . In our context, N^{shot} is usually a small integer (hence the term *few-shot*). Consequently, these tasks aim to mimic the realistic situation in low data regimes. We denote the dataset for the ℓ -th environment by $\mathcal{D}_\ell$. Each $\mathcal{D}_\ell$ is split into a *support set* $\mathcal{D}_\ell^{\text{S}}$ for learning and a *query set* $\mathcal{D}_\ell^{\text{Q}}$ for training and testing. We drop the subscript ℓ in the following discussions as the learning procedure is identical in all environments.

Meta-learning is often implemented by learning over a distribution of tasks $P(\mathcal{T})$ rather than individual data distribution. Its training phase can then be formulated as follows:

$$\theta^*, \xi^{\text{meta}*} = \arg \min_{\theta, \xi^{\text{meta}}} \mathbb{E}_{\mathcal{T} \sim P(\mathcal{T})} \mathcal{L}(\mathcal{D}; \theta; \xi^{\text{meta}}). \quad (4.4)$$

To find the optimal $\xi^{\text{meta}*}$ in Eqn. (4.4), meta-learning employs a set of source tasks, then it iteratively optimizes θ and ξ^{meta} (over $\mathcal{D}^S$ and $\mathcal{D}^Q$ respectively) as follows:

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{\mathcal{T} \sim P(\mathcal{T})} \mathcal{L}^{\text{task}}(\mathcal{D}^S; \xi^{\text{meta}*}; \theta), \quad (4.5)$$

$$\xi^{\text{meta}*} = \arg \min_{\xi^{\text{meta}}} \mathbb{E}_{\mathcal{T} \sim P(\mathcal{T})} \mathcal{L}^{\text{meta}}(\mathcal{D}^Q; \xi^{\text{meta}}; \theta^*), \quad (4.6)$$

where $\mathcal{L}^{\text{task}}$ and $\mathcal{L}^{\text{meta}}$ denote the task and meta objectives respectively. To be specific, $\mathcal{L}^{\text{task}}$ searches for optimal θ^* conditioning on meta knowledge ξ^{meta} that computed from achieving meta objective. In a nutshell, the meta objective trains meta knowledge ξ^{meta} in such a way that it produces $f_{\theta^*, \xi^{\text{meta}}}$ which performs well on their query sets. Upon a new target task with dataset $\hat{\mathcal{D}}$ (containing $\hat{\mathcal{D}}^S$ as the support set with a minimum level of labeled data, and the rest unlabeled for final testing), $\xi^{\text{meta}*}$ is transferred and f_{θ} is refined according to $\hat{\mathcal{D}}^S$.

Different forms of ξ^{meta} could be taken, including the initial values of θ [143], optimization strategy [144], embedding [145], and hyperparameter [146]. Meta-learning is categorized into optimization-based, model-based, and metric-based. Since learning an embedding function is often deemed suitable to deal with few-shot applications [145, 147], we choose embedding as ξ^{meta} for our design (i.e., following the metric-based framework [145]) as it is much more efficient than others [148]. Moreover, MetaSense [36], the only work utilizing meta-learning in this context, employ optimization-based meta-learning. Therefore, we mainly detail optimization-based meta-learning and metric-based meta-learning in the following sections.

4.1.2.1 Optimization-Based

This research line focuses on extracting meta knowledge to improve task objective optimization performance. Most of them can be employed to cope with few labeled data samples or to converge within a few optimization steps. Meta knowledge can be weight initialization and optimization steps, etc.

In the first family of methods, training algorithms are adjusted to find good initialization weights θ_0 as ξ^{meta} , so that f_{θ_0} could be generalized to new tasks with few labeled

data samples or converge within a few optimization steps. The most famous algorithm is Model-Agnostic Meta-Learning (MAML) [143]. It claims to be applicable to any network learned via gradient descent. Formally, for each $\mathcal{T}_i \sim P(\mathcal{T})$, two procedures are conducted: i) $\nabla_{\theta} \mathcal{L}^{\mathcal{T}_i}(f_{\theta})$ with respect to support samples is computed and ii) adapted parameter with gradient descent is calculated as $\theta'_i = \theta - \nu^{\text{task}} \nabla_{\theta} \mathcal{L}^{\mathcal{T}_i}(f_{\theta})$, where ν^{task} is step size. Afterward, all the adapted parameters $\{\theta_i\}$ with unseen query samples in D_i are employed for meta objective optimization as follows:

$$\theta \leftarrow \theta - \nu^{\text{meta}} \nabla_{\theta} \sum_{\mathcal{T}_i \sim P(\mathcal{T})} \mathcal{L}^{\mathcal{T}_i}(f_{\theta'_i}), \quad (4.7)$$

where ν^{meta} is meta step size.

Recently, MetaSense [36] directly adopts MAML to construct an adaptive wearable sensing system. It learns how to adapt to the target environments by rehearsing related source environments. Consequently, its system can be fast adapted to the target environments with few labeled samples. A key challenge with this approach is that meta optimization needs to solve for as many parameters as the task optimization (potentially hundreds of millions in large CNNs). This leads to a line of work on improving learning efficiency. Later proposals (e.g., Reptile [149], TAML [150]) all intend to improve learning efficiency along various directions.

The aforementioned algorithms solve optimization with existing optimizers such as Adam [151], etc. The second family of algorithms intends to acquire optimization steps via training rather than hand-designed [152, 153]. Compared with weight initialization, parameters to learn here can potentially be few if the optimizer is applied coordinated-wise across weights [154].

4.1.2.2 Metric-Based

This category of approaches intends to classify a new sample $\mathbf{x}$ by a weighted sum of the labels in $\mathcal{D}^S$ as follows:

$$P(y|\mathbf{x}, \mathcal{D}^S) = \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}^S} \mathcal{M}(h_{\theta}(\mathbf{x}), h_{\theta}(\mathbf{x}_i)) y_i, \quad (4.8)$$

where $\mathcal{M}$ is an optimal metric function for measuring the similarity between $\mathbf{x}$ and $\mathbf{x}_i$. Essentially, they aim to learn an embedding function h_{θ} that transforms inputs into a

representation suitable for classification via the similarity comparison. Metrics $\mathcal{M}$ such as Cosine similarity or Euclidean distance are commonly used for similarity comparison. Although several proposals have been made under this category, they mostly differ in the choice of the (input) embedding vectors and distance metric.

Kosh *et al.* [155] propose Siamese network to solve one-shot image recognition. The assumption is that the learned embedding can be employed to measure the distance between images of unknown categories. To be specific, Siamese network utilizes h_θ to encode $\mathbf{x}_i$ and $\mathbf{x}_j$ and then l_1 distance of two embeddings $\|h_\theta(\mathbf{x}_i) - h_\theta(\mathbf{x}_j)\|$ is converted to the probability $P^{\text{same}}(\mathbf{x}_i, \mathbf{x}_j)$ of whether $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class. After acquiring optimal h_θ in source tasks, given $\mathcal{D}^S$ and query image $\mathbf{x}$ in target task, the final predicted class is computed as follows:

$$\hat{y} = c(\arg \max_{\mathbf{x}_i \in \mathcal{D}^S} P^{\text{same}}(\mathbf{x}, \mathbf{x}_i)), \quad (4.9)$$

where c denotes the class label.

Vinyals *et al.* [145] design matching network to perform one-shot image recognition. Essentially, final prediction of query image $\mathbf{x}$ is defined as a sum of labels of $\mathcal{D}^S$ weighted by attention kernel $a_\theta(\mathbf{x}, \mathbf{x}_i)$ that trained in source tasks. $a_\theta(\mathbf{x}, \mathbf{x}_i)$ is defined as follows:

$$a_\theta(\mathbf{x}, \mathbf{x}_i) = \frac{\exp(\cosine(h_\theta(\mathbf{x}), h_\theta(\mathbf{x}_i)))}{\sum_{j=1}^k \exp(\cosine(h_\theta(\mathbf{x}), h_\theta(\mathbf{x}_j)))}, \quad (4.10)$$

where k denotes the number of labeled samples in support set.

Consequently, the final prediction probability is defined as follows:

$$\hat{y} = \sum_{i=1}^k a_\theta(\mathbf{x}, \mathbf{x}_i) y_i. \quad (4.11)$$

Snell *et al.* [147] propose prototypical network for solving image recognition. Similarly, it employs h_θ to encode images into feature representations. Differently, prototypical network predicts query image $\mathbf{x}$ by comparing with *prototype* rather than all the support samples in $\mathcal{D}^S$. Prototype v_c is defined for class $c \in N^{\text{class}}$ and formulated as the mean vector of the embedded support data samples in a certain class. Consequently, the prediction probability of query image x in the target domain is defined as follows:

$$\hat{y} = \sigma_x(-\mathcal{M}(h_\theta(\mathbf{x}), v_c)), \quad (4.12)$$

where $\mathcal{M}$ is performed by squared Euclidean distance.

Table 4.1: Time required to solve 10,000 miniImageNet episodes of 10 samples each.

Approaches	$N^{\text{class}} = 5, N^{\text{shot}} = 1$
MAML	6 min 35 sec
Prototypical Network	27 sec

4.1.2.3 Drawbacks

Although MetaSense [36] employs optimization-based meta-learning to perform environment adaptation under data scarcity, it is only deployed to wearable sensing context instead of device-free RF sensing environment. Moreover, the generalization ability of optimization-based meta-learning used in MetaSense is questionable. [37] states that using only a few samples to compute gradients in a high-dimensional parameter space could make generalization difficult. Furthermore, as shown in Table 4.1 (results are provided in [148]), when MAML and prototypical networks have similar capacity, we could observe that MAML is a costly iterative approach compared to prototypical network. The reason is that meta objective optimization needs to solve for as many parameters as the task objective optimization. Therefore, when it is applied to f_θ with a high-dimensional parameter space, tuning initialization weights could result in generalization difficulty given very few samples. Compared with RF sensing data, the dimension of wearable sensing data obtained by IMUs is much lower. Therefore, they can adopt a shallow neural network as f_θ to avoid the generalization challenge described above. However, RF sensing certainly demands a powerful base network f_θ (with a high-dimensional parameter space for ξ^{meta}), rendering optimization-based meta-learning approaches largely inapplicable.

4.1.3 Our Proposal

As discussed above, existing solutions have the following drawbacks. Each time transfer learning based approaches [25, 35] has to be performed again with a substantial amount of labeled data when they are applied to a new target environment. Optimization-based meta-learning approach [36] is only deployed to device-based sensing context instead of device-free RF sensing environment. Moreover, their generalization ability and efficiency are questionable. Compared with optimization-based meta-learning approaches, metric-based approaches are rather straightforward: they mainly focus on directly learning a

powerful embedding function that transforms inputs into a representation suitable for classification via the similarity comparison, rather than duplicate tuning f_θ . This property has made metric-based meta-learning approaches less constrained by the complexity of f_θ , and it has also motivated us to adopt metric-based approaches for achieving RF-HAR environment adaptation under data scarcity. Moreover, as explained in Sec. 2.2.3, existing RF-HAR systems only utilize partial information provided by RF signals. It is also motivated us to design a unified framework to handle representative RF sensing techniques described in Sec. 2.2.

Consequently, leveraging the power of metric-based meta-learning, we propose RF-Net for one-shot RF-HAR, which applies to all representative RF sensing techniques. It performs HAR accurately in a new environment with only one sample for each label. We innovate in two aspects: i) a unified dual-path base network for extracting high-dimensional features contained in the signal matrix (thus applicable to all three RF sensing datasets mentioned in Sec. 2.2), and ii) a metric-based meta-learning framework learning from seen environments, where enables base HAR network that transforms inputs into a representation suitable for classification via the similarity comparison and then allows environment adaptation with very few labeled data.

For the base HAR network, we combine spatial module along with attention-based temporal module, aiming to learn influential spatial-temporal features from both time and frequency domains. Our meta-learning framework contains a parametric RF-specific module designed to train a powerful distance metric instead of simply relying on traditional non-parametric metrics (e.g., Euclidean or Cosine distances). This helps to achieve a better generalization when applying this metric to conduct classifications in new environments. Essentially, we employ the base network to perform both activity recognition and feature extraction and exploit them to (meta)-train the distance metric via a residual classification module.

4.2 RF-Net: One-Shot HAR

In this section, we first describe the problem formulation. Then we elaborate our meta-learning framework and dual-path base HAR network.

4.2.1 Problem Formulation

In this paper, the ultimate goal of RF-Net q_{Ω} , parameterized by Ω , is to perform one-shot RF-HAR, i.e., adapting to every new environment rapidly with a single labeled sample per class. To achieve it, we need a base HAR network f_{Φ} , parameterized by Φ , to extract features from input samples (i.e., RF signal matrix). We deem an RF signal matrix $\mathbf{x} \in \mathbb{R}^{K \times L \times N^r}$ as an image, where K is slow time dimension, L is fast time dimension, and N^r is the number of transmitter-receiver (tx-rx) pairs. This base network f_{Φ} is then wrapped into a meta-learning framework to be generalized to new environments. In order to maintain the generalization capability of f_{Φ} given its high-dimensional parameter space, we adopt a metric-based meta-learning framework g_{Θ} parameterized by Θ . Therefore, RF-Net q_{Ω} includes the base network f_{Φ} and meta-learning framework g_{Θ} , hence $\Omega = \Phi \cup \Theta$.

The procedure for learning q_{Ω} is planned as follows. We first train RF-Net with N^e source environment training datasets $\mathcal{D} = \{\mathcal{D}_{\ell}\}_{\ell=1}^{N^e}$. For each epoch, we train q_{Ω} on environment datasets, learning from environment to environment to mimic how RF-Net would be tested when presenting in a new environment. More specifically, for each environment dataset $\mathcal{D}_{\ell}$, we sample support samples as support set $\mathcal{D}_{\ell}^S = \{\mathbf{x}_{\ell,j}^S, y_{\ell,j}^S\}_{j=1}^{N^{\text{class}}}$, $\mathbf{x}_{\ell,j}^S \in \mathbb{R}^{N^{\text{shot}} \times K \times L \times N^r}$ and query samples as query set $\mathcal{D}_{\ell}^Q = \{\mathbf{x}_{\ell}^Q, y_{\ell}^Q\}$, $\mathbf{x}_{\ell}^Q \in \mathbb{R}^{1 \times K \times L \times N^r}$, where N^{shot} denotes the number of samples (i.e., $N^{\text{shot}} = 1$ for *one-shot* learning), N^{class} is the number of activity categories, and y_{ℓ} denotes activity label. Note that the support set $\mathcal{D}_{\ell}^S$ and query set $\mathcal{D}_{\ell}^Q$ belong to the same environment space but samples are disjoint, i.e., $\mathcal{D}_{\ell}^S \cap \mathcal{D}_{\ell}^Q = \emptyset$. We drop the subscript ℓ as the learning procedure is identical in all environments. Essentially, the objective function of RF-Net q_{Ω} can be formulated as follows:

$$\Omega^* = \arg \max_{\Omega} \mathbb{E}_{\mathcal{D}} \left[\sum_{\mathcal{D}^Q} \sum_{\mathcal{D}^S} g_{\Theta} \left(f_{\Phi}(\mathbf{x}^Q), f_{\Phi}(\mathbf{x}^S), y^S \right) \right]. \quad (4.13)$$

 Figure 4.1: Overall design of RF-Net q_{Ω} .

4.2.2 Meta Framework

We intend to design a trainable metric-based meta-learning framework specifically for RF signal matrices. As illustrated in Figure 4.1, our proposed meta framework g_{Θ} consists of two modules:

- RF Metric Module g^{metric} : it aims to train a powerful distance metric for measuring sample similarities via a linear mapping layer. Meanwhile, it takes into consideration the multiple features generated by the dual-path base network f_{Φ} . Consequently, this module could provide a holistic interpretation of $\mathbf{x}$ toward better generalization.
- Residual Classification Module g^c : it intends to incorporate the capability of f_{Φ} further as a recognition network, in addition to its feature extractor role in g^{metric} . This module allows the base network to assist in (meta)-training the distance metric g^{metric} .

4.2.2.1 RF Metric Module

RF metric module first employs our dual-path base HAR network (see Section 4.2.3) as a (deep) feature extractor $f_\Phi : \mathbb{R}^{D_x} \rightarrow \mathbb{R}^{D_z}$, mapping an RF signal matrix $\mathbf{x}$ to a feature representation $\mathbf{z}$, where D_x and D_z are the dimensionalities of RF signal matrix and the corresponding embedding, respectively. Different from previous works such as [25, 72, 145], we expect that the metric space of multiple features is able to capture a more complex representation than simply relying on single feature representation. Therefore, considering the RF signal matrix described in both time and frequency domains, we leverage M features to represent the RF signal matrix $\mathbf{x}$ as follows:

$$\{H_m^{\text{feat}} = f_\Phi(\mathbf{x})\}_{m=1}^M.$$

In our problem setting, H_m^{feat} includes H^{time} and H^{freq} respectively extracted from time and frequency domains, as well as H^{fuse} combining features from both domains to derive spatial and temporal information. We refer to Section 4.2.3 for further elaborations on these features. Given these features, we employ cosine distance $\mathcal{M}(a, b) = -\frac{a \cdot b}{\|a\| \|b\|}$ to obtain a distance set $\{\lambda_m\}_{m=1}^M$ between features generated by support samples and query samples as follows:

$$\{\lambda_m\}_{m=1}^M = \left\{ \mathcal{M}\left(H_m^{\text{feat}, S}, H_m^{\text{feat}, Q}\right) \right\}_{m=1}^M.$$

In order to further increase the RF metric space, we design RF metric module to be trainable instead of solely relying on cosine distance. We stack these M distances $[\lambda_1, \lambda_2, \dots, \lambda_M]$ into a vector form $\mathbf{\Lambda}$. Given a query sample, we propose to combine each distance measure λ_m in $\mathbf{\Lambda}$ via learnable linear mapping weights η . Therefore, we could compute the activity probabilities of query samples by weighting the labels of support samples y^S as follows:

$$P_\beta(y = j | \mathbf{x}) = \sigma_x(-\Lambda_j \beta), \beta = \eta y^S, \quad (4.14)$$

where $\eta \in \mathbb{R}^{M \times N^{\text{class}}}$, σ_x denotes Softmax function, and $\Lambda_j \in \mathbb{R}^{1 \times M}$.

To further realize the role of β , we analyze that for j -th class, the class-wise cross-entropy loss function is given by

$$\mathcal{L}_j(\beta) = \sum_{\mathbf{x} \in \mathcal{D}^Q} \left[\Lambda_j \beta + \log \left(\sum_{n=1}^{N^{\text{class}}} e^{-\Lambda_n \beta} \right) \right]. \quad (4.15)$$

For each sample of query set $\mathbf{x} \in \mathcal{D}^Q$, the second-order partial derivative of Eq (4.15) with respect to β is

$$\nabla^2 \mathcal{L}_j(\beta) = \frac{1}{\mathbf{1}^T \gamma} \text{diag}(\gamma) - \frac{1}{(\mathbf{1}^T \gamma)^2} \gamma \gamma^T, \quad (4.16)$$

where $\gamma = [\gamma_1, \dots, \gamma_{N^{\text{class}}}]$ and $\gamma_k = e^{-\Lambda_k \beta}$. If $\nabla^2 \mathcal{L}_j(\beta) \geq 0$, $\mathcal{L}_j(\beta)$ in Eq (4.15) is convex.

Therefore, we need to verify that $\mathbf{v}^T \nabla^2 \mathcal{L}_j(\beta) \mathbf{v} \geq 0$ for all $\mathbf{v}$, but we have:

$$\mathbf{v}^T \nabla^2 \mathcal{L}_j(\beta) \mathbf{v} = \frac{(\sum_{n=1}^{N^{\text{class}}} \gamma_n v_n^2)(\sum_{n=1}^{N^{\text{class}}} \gamma_n) - (\sum_{n=1}^{N^{\text{class}}} v_n \gamma_n)^2}{(\sum_{n=1}^{N^{\text{class}}} \gamma_n)^2},$$

which is indeed non-negative due to Cauchy-Schwarz inequality, i.e., $(\sum_{n=1}^{N^{\text{class}}} v_n \gamma_n)^2 \leq (\sum_{n=1}^{N^{\text{class}}} \gamma_n v_n^2)(\sum_{n=1}^{N^{\text{class}}} \gamma_n)$. Now we have proven the convexity of $\mathcal{L}_j(\beta)$, which in turn indicates that our RF metric module can quickly learn the optimal parameters β for combining multiple representations together.

4.2.2.2 Residual Classification Module

In this module, different from traditional metric-based meta-learning framework, we aim to employ the base network f_Φ for recognition too, rather than solely as a feature extractor explained in Section 4.2.2.1. To be specific, we first compute $\hat{\mathbf{y}}^{\text{meta},Q}$, i.e., classify query samples by weighting the labels of support sample $\mathbf{y}^S$, as described in Eq (4.14). Meanwhile, we exploit f_Φ as recognition network to compute logits directly $\hat{\mathbf{y}}^Q = f_\Phi(\mathbf{x}^Q)$. Afterwards, we incorporate $\hat{\mathbf{y}}^Q$ into $\hat{\mathbf{y}}^{\text{meta},Q}$ via a residual connection, so that the final predicted logits of RF-Net $\tilde{\mathbf{y}}^Q$ is computed as $g^c(g^{\text{metric}}(f_\Phi(\mathbf{x}^S), f_\Phi(\mathbf{x}^Q)), \mathbf{y}^S, \hat{\mathbf{y}}^Q) = g^{\text{metric}}(f_\Phi(\mathbf{x}^S), f_\Phi(\mathbf{x}^Q))\mathbf{y}^S + \hat{\mathbf{y}}^Q$. It enables the base network to reinforce the meta-training of the distance metric g^{metric} .

4.2.2.3 Training Strategy

We carefully devise our training strategy for learning RF-Net. As described in **Algorithm 2**, given RF-Net q_Ω including the base network f_Φ and meta-learning framework g_Θ , we exploit training dataset $\mathcal{D}$ and testing dataset $\hat{\mathcal{D}}$ in training and testing stages, respectively.

Algorithm 2 RF-Net training.

Input: Training dataset $\mathcal{D}$, Testing dataset $\hat{\mathcal{D}}$, Base network f_{Φ} , RF-Net q_{Ω} , Meta-learning framework g_{Θ} , hyperparameters α and β , as well as the maximum number of iterations iter

```

1: % Training
2: while iter  $\neq 0$  do
3:   Sample environment minibatch  $\{\mathcal{D}_{\ell}\} \sim \mathcal{D}$ 
4:   for  $\mathcal{D}_{\ell} \in \{\mathcal{D}_{\ell}\}$  do
5:     Sample support and query samples  $\mathcal{D}_{\ell}^S, \mathcal{D}_{\ell}^Q \sim \mathcal{D}_{\ell}$ 
6:      $\mathcal{D}_{\ell}^S = \{\mathbf{x}_{\ell,j}^S, y_{\ell,j}^S\}_{j=1}^{N^{\text{class}}}$ ,  $\mathcal{D}_{\ell}^Q = \{\mathbf{x}_{\ell}^Q, y_{\ell}^Q\}$ ,  $\mathcal{D}_{\ell}^Q \cap \mathcal{D}_{\ell}^S = \emptyset$ 
7:     Evaluate  $\nabla_{\Omega} \mathcal{L}(f_{\Phi}(\mathbf{x}_{\ell}^S), y_{\ell}^S)$ ,  $\Omega \leftarrow \Omega - \alpha \nabla_{\Omega} \mathcal{L}_{\ell}$ 
8:     Evaluate  $\nabla_{\Theta} \mathcal{L}(q_{\Omega}(\mathbf{x}_{\ell}^S, \mathbf{x}_{\ell}^Q, y_{\ell}^S), y_{\ell}^Q)$ ,  $\Theta \leftarrow \Theta - \beta \nabla_{\Theta} \mathcal{L}_{\ell}$ 
9:   end for
10:  iter  $\leftarrow$  iter  $- 1$ 
11: end while
12: % Testing
13: for  $\hat{\mathcal{D}}_{\ell} \in \hat{\mathcal{D}}$  do
14:  Sample support and query samples  $\hat{\mathcal{D}}_{\ell}^S, \hat{\mathcal{D}}_{\ell}^Q \sim \hat{\mathcal{D}}_{\ell}$ 
15:   $\hat{\mathcal{D}}_{\ell}^S = \{\mathbf{x}_{\ell,j}^S, y_{\ell,j}^S\}_{j=1}^{N^{\text{class}}}$ ,  $\hat{\mathcal{D}}_{\ell}^Q = \{\mathbf{x}_{\ell}^Q, y_{\ell}^Q\}$ ,  $\hat{\mathcal{D}}_{\ell}^Q \cap \hat{\mathcal{D}}_{\ell}^S = \emptyset$ 
16:  Evaluate  $\nabla_{\Theta} \mathcal{L}(f_{\Phi}(\mathbf{x}_{\ell}^S), y_{\ell}^S)$ ,  $\Theta \leftarrow \Theta - \alpha \nabla_{\Theta} \mathcal{L}_{\ell}$ 
17:  Predict  $\hat{y}^Q = q_{\Omega}(\mathbf{x}_{\ell}^S, \mathbf{x}_{\ell}^Q, y_{\ell}^S)$ 
18: end for
    
```

In training stage (line 2-11), we train f_{Φ} and g_{Θ} in the leader-follower asymmetric manner between inner- and meta-training to enhance the overall capability on performing one-shot RF-HAR with q_{Ω} . More specifically, we first inner train f_{Φ} with $\mathcal{D}^S$ (line 7). We minimize the inner training loss $\mathcal{L}(f_{\Phi}(\mathbf{x}^S), y^S)$ by conducting gradient descent with respect to both base network parameters Φ and meta parameters Θ . This helps finding a coarse initialization points for learning meta parameters Θ , boosting the convergence speed of RF-Net q_{Ω} . Afterwards (line 8), we meta-train g_{Θ} conditioned on learned f_{Φ} using both support and query samples, so that it produces q_{Ω} that performs well on recognizing query samples.

Now we have achieved a well-trained RF-HAR q_{Ω} by sequentially training f_{Φ} and g_{Θ} . Given a testing dataset $\hat{\mathcal{D}}$, we could evaluate the performance of RF-Net on performing one-shot RF-HAR. We first intend to refine q_{Ω} with labeled support samples $\hat{\mathcal{D}}^S = \{\mathbf{x}_j^S, y_j^S\}_{j=1}^{N^{\text{class}}}$. To be specific, we adapt the well-trained q_{Ω} using the same inner training

 Figure 4.2: Overall design of dual-path base network f_{Φ} .

procedure, except that we only fine-tune meta-learning framework parameter Θ (lines 16). Finally, we classify query samples with refined q_{Ω} and labeled support samples as $\hat{y}^Q = q_{\Omega}(\mathbf{x}^S, \mathbf{x}^Q, \mathbf{y}^S)$.

4.2.3 Dual-Path Base Network

As illustrated in Figure 4.2, our base HAR network f_{Φ} is composed of three main modules to learn RF feature representations:

- Spatial Module f^{spatial} : it extracts sensitive spatial features from RF matrices $\mathbf{x}$ in both time and frequency domains.
- Attention-based Temporal Module f^{temporal} : it aims to capture long-term temporal features from RF matrices in both time and frequency domains. And it also intends to generate joint temporal representations across two domains.
- Classification Module $f^{\text{classification}}$: this final module predicts activity label, given trained features from f^{spatial} and f^{temporal} .

For both f^{spatial} and f^{temporal} , we first compute $\mathbf{x}_f \in \mathbb{R}^{K \times L \times N^f}$ via FFT along the slow time of an RF matrix $\mathbf{x} \in \mathbb{R}^{K \times L \times N^t}$. Then both $\mathbf{x}$ (time domain) and $\mathbf{x}_f$ (frequency domain) are used as input.

Figure 4.3: Two alternative spatial modules.

4.2.3.1 Spatial Module

This module leverages f^{spatial} to learn the spatial features H^{spat} . Essentially, it extracts sensitive spatial features from RF matrices in both time domain $\mathbf{x}$ and frequency domain $\mathbf{x}_f$. We employ state-of-the-art deep learning model CNNs as the backbone f^{backbone} , and we empirically select the most cost-effective backbone in Section 4.3.3.1. Generally, CNNs are designed for exploring spatial features in image data. Therefore, we regard $\mathbf{x}$ and $\mathbf{x}_f$ as image with N^t tx-rx pair as input channels, L fast time dimension as height of input planes in pixels and K slow time dimension as width in pixels. Moreover, we additionally employ one convolutional layer f^{conv} to adjust channel dimension into designated dimension declared by the backbone.

We explore two spatial modules, named spatial separate module and spatial fuse module. As depicted in Figure 4.3a, spatial separate module aims to employ backbone for generating spatial features of $\mathbf{x}$ and $\mathbf{x}_f$ separately: $H^{\text{s},\mathbf{x}} = f^{\text{backbone}}(f^{\text{conv}}(\mathbf{x}))$ and $H^{\text{s},\mathbf{x}_f} = f^{\text{backbone}}(f^{\text{conv}}(\mathbf{x}_f))$; the final spatial features H^{spat} are then produced by concatenating $H^{\text{s},\mathbf{x}}$ and $H^{\text{s},\mathbf{x}_f}$. However, spatial separate module involves a high computation complexity due to repeated backbone usage. Therefore, we further explore spatial fuse module illustrated in Figure 4.3b. As a simplification, we concatenate $\mathbf{x}$ and $\mathbf{x}_f$ at the initial stage $\mathbf{x}_c = [\mathbf{x}, \mathbf{x}_f] \in \mathbb{R}^{K \times L \times 2N^t}$. Afterwards, we intend to learn a composite representation from $\mathbf{x}_c$. We first reshape $\mathbf{x}_c \in \mathbb{R}^{K \times 2N^t L}$ and then we employ a Rectified Linear Unit (ReLU) activated dense layer to generate a composite representation $H^c \in \mathbb{R}^{K \times \alpha \times 2}$, where α denotes the hidden dimension. Afterwards, we employ backbone f^{backbone} to

Figure 4.4: Attention-based temporal module.

directly extract $H^{\text{spat}} = f^{\text{backbone}}(f^{\text{conv}}(H^c))$. The corresponding performance comparisons between these two modules, along with the choice of α and activation function, are conducted in Section 4.3.

4.2.3.2 Attention-Based Temporal Module

In order to generate long-term temporal features from RF matrices in both time and frequency domains, we utilize f^{temporal} to learn attended time features H^{time} and attended frequency features H^{freq} . As illustrated in Figure 4.4, we first extract initial time features $H^{\text{time}\Diamond}$ and frequency features $H^{\text{freq}\Diamond}$, leveraging Long Short Term Memory (LSTM) to avoid gradient vanishing problems. Therefore, we reshape $\mathbf{x} \in \mathbb{R}^{K \times (L \times N^t)}$ and $\mathbf{x}_f \in \mathbb{R}^{K \times (L \times N^f)}$, regarding the slow time dimension as step index with cardinality K . Then, we extract $H^{\text{time}\Diamond} \in \mathbb{R}^{K \times \alpha}$ and $H^{\text{freq}\Diamond} \in \mathbb{R}^{K \times \alpha}$ by passing $\mathbf{x}$ and $\mathbf{x}_f$ into LSTM separately. After that, we focus on employing attention mechanism [156] to learn richer joint representations between initial feature representations, aiming to generate the attention map $\mathcal{A} \in \mathbb{R}^{K \times K}$ containing the joint information of every step between $H^{\text{time}\Diamond}$ and $H^{\text{freq}\Diamond}$. To be specific, ReLU activated dense layers are first employed to map $H^{\text{time}\Diamond}$, $H^{\text{freq}\Diamond}$ into $H^{\text{time}\dagger}$, $H^{\text{freq}\dagger} \in \mathbb{R}^{K \times \iota}$, where ι denotes hidden dimension. Then the attention

map is generated as $\mathcal{A} := \text{softmax}(W \circ H^{\text{time}\dagger} (H^{\text{freq}\dagger})^T)$, where $\circ$ stands for Hadamard product, T denotes transpose, and $W \in \mathbb{R}^{K \times \iota}$ denotes weight matrix.

Combing the attention map $\mathcal{A}$ with the initial features produces joint representations $((H^{\text{time}\diamond})^T \mathcal{A})^T$ and $((H^{\text{freq}\diamond})^T \mathcal{A})^T$, and these joint representations are passed through ReLU activated dense layers for adding non-linearity. Especially, we separately incorporate these mapped joint representations into $H^{\text{time}\diamond}$, $H^{\text{freq}\diamond}$ via a residual connection to generate $H^{\text{time}} \in \mathbb{R}^{K \times \alpha}$, $H^{\text{freq}} \in \mathbb{R}^{K \times \alpha}$, thus it is able to solve vanishing gradients issue and achieve optimal identity mapping [157]. Subsequently, we choose final step (K -th) of attended time and frequency features, i.e., H_K^{time} and H_K^{freq} , for producing a composed temporal features H^{temp} . To be specific, we first employ two ReLU activated dense layers to generate H_K^{time} and H_K^{freq} . Then, we stack those mapped features followed by a dense layer to compute $H^{\text{temp}} \in \mathbb{R}^{1 \times 2\alpha}$.

4.2.3.3 Classification Module

We have employed spatial module f^{spatial} and attention-based temporal module f^{temporal} to generate sensitive spatial features H^{spat} and attended temporal features H^{temp} from both time and frequency domains. In order to efficiently predict an activity label, we further derive an integrated feature by fusing both spatial and temporal representations. In this paper, we directly add spatial features H^{spat} and temporal features H^{temp} as $H^{\text{fuse}} = H^{\text{temp}} + H^{\text{spat}}$. To enable this integration, we set the hidden dimension α properly so as to equalize the dimensions of H^{spat} and H^{temp} . Finally, we utilize a dense layer to predict activity label $\hat{y} = H^{\text{fuse}} W_1$, where $W_1 \in \mathbb{R}^{2\alpha \times N^{\text{class}}}$ is weight matrix.

Remark: We choose the final step (K -th) of the attended temporal features H_K^{time} and H_K^{freq} , along with the fused features H^{fuse} , as $\{H_m^{\text{feat}}\}_{m=1}^M$ in RF-Net (see Section 4.2.2.1).

4.3 Experiments

In this section, we conduct extensive experiments on all three RF sensing techniques, i.e., Wi-Fi, Frequency-Modulated Continuous Wave (FMCW), and impulse radio (IR), aiming to demonstrate the efficacy of the dual-path base network for activity recognition and to evaluate the overall performance of RF-Net on performing one-shot RF-HAR. In

Table 4.2: Datasets information.

Sensing	Environments #	samples #	Activities #
Wi-Fi	80	25	6
Wi-Fi	100	20	6
Wi-Fi	120	16	6
FMCW	10	17	6
IR	50	16	6

particular, we report experiments on the following aspects: i) hyperparameter searching of network setting, ii) superiority of RF-Net and dual-path base network over baselines, and iii) efficacy of RF-Net and dual-path base network on various RF sensing datasets.

4.3.1 Datasets

RF signal in each sensing technique could be formulated as matrix $\mathbf{x} \in \mathbb{R}^{K \times L \times N'}$. In this section, we first elaborate RF signal matrix of these RF sensing techniques. Then we describe environment information and human activities provided in each dataset. Given the variety of testing conditions, the activities of testing subjects may vary across different datasets.

Details of each dataset are summarized in Table 4.2: for each RF sensing technology, we specify the number of environments involved, the number of samples per environment taken, and the number of activities tested. To generate different environments, we first select a few rooms with distinct sizes, also involve different testing subjects. Within the same room, we change the positions of the furniture and appliances and the location of the subject. In order to artificially create “differences”, we make sure that at least five objects (including the subject) have their position changed in a room when generating a new environment.

Due to the popularity of Wi-Fi sensing, we choose to employ three datasets for it, but only one dataset for FMCW and IR.

4.3.1.1 Wi-Fi

We exploit Channel State Information (CSI) of 30 Orthogonal Frequency Division Multiplexing (OFDM) subcarriers with two tx-rx pairs to record six human activities: wiping,

walking, moving, rotating, sitting, and standing up. CSI information is sampled at 100Hz and conducted window slicing size in 5.12s. Therefore, each signal matrix is with a total $K = 512$ received packets, $L = 30$ subcarriers, and $N^r = 2$ tx-rx pairs. We employ 11 subjects and record from 6 different rooms. In the following experiments, we intend to investigate the impact of the number of environments as well. Therefore, we split these environments data into three datasets, including 80 environments, 100 environments, and 120 environments. As described in Table 4.2, these datasets have 25, 20, and 16 samples per environment per activity, respectively.

4.3.1.2 FMCW

We utilize an FMCW radar device with one tx-rx pair to collect data. To be specific, we collect each frame with 253 frequency components every 67 ms. Then, we stack 100 frames for covering a 6.7 s interval. Therefore, each FMCW matrix $\mathbf{x} \in \mathbb{R}^{K \times L \times N^r}$ is with a total $K = 100$ frames, $L = 253$ frequency components, and $N_r = 1$ tx-rx pair. These matrices depict that nine subjects in 2 different rooms perform six activities: standing up, sitting down, going out, entering the room, putting on clothes, and putting off clothes. We aim to keep consistency on the number of samples per environment per activity of all datasets. Therefore, we split samples into 10 different environments.

4.3.1.3 IR

We employ an IR device with one tx-rx pair to transmit the pulse signal. Specifically, we collect each frame with 138 time components every 2.5 ms, and we stack 400 frames together. Accordingly, each IR matrix $\mathbf{x} \in \mathbb{R}^{K \times L \times N^r}$ is with total $K = 400$ frames, $L = 138$ samples of a pulse, and $N^r = 1$ tx-rx pair. We employ 20 subjects in 3 rooms to perform six activities: sitting down, standing up, walking, falling, bending, and lying. We split all the samples into 50 different environments.

4.3.2 Baselines & Implementation

We present several baseline schemes against which our RF-Net and its base network will be compared. We also briefly explain the implementation of our experiments.

4.3.2.1 Baselines for Dual-Path Base Network

We compare our proposed base network with the following baseline networks. We aim to demonstrate the efficacy of our base network on extracting features from the RF signal matrix for HAR. Following networks all employ RF metric module for dealing with data scarcity.

- TIME: the proposed spatial module with only time domain data. It is used to evaluate network with spatial features in time domain.
- FREQ: the proposed spatial module with only frequency domain data. It is used to investigate the performance of network with spatial features in frequency domain.
- SPATIAL-SEP: the proposed spatial separate module. It is used to investigate the performance of network with spatial features in both time and frequency domain.
- SPATIAL-FUSE: the proposed spatial fuse module. It is used to examine network with spatial features in both time and frequency domain.
- DUAL-wo-ATT: the proposed dual-path base network without attention mechanism. It is used to demonstrate the efficacy of the attention mechanism.
- DUAL-PATH: the proposed dual-path base network.

Note that, as SPATIAL-SEP and SPATIAL-FUSE are two alternative components, DUAL-wo-ATT and DUAL-PATH will choose to involve whichever performs better.

4.3.2.2 Baselines for RF-Net

We evaluate the performance of one-shot human activity recognition when RF-Net is adapted to new environments. Therefore, we mainly compare our network with the following knowledge transfer approaches: meta-learning baselines and fine-tuning baseline. Following networks all employ DUAL-PATH as base network.

- Fine-Tuning (FT): FT firstly trains dual-path base network with the training set. When FT adapts the network on the test set, the parameters of classification module would be fine-tuned with support samples on the test set.

- Model-Agnostic Meta-Learning (MAML): a state-of-the-art optimization-based meta-learning baseline. This scheme has been adopted by MetaSense [36] to enable adaptive wearable sensing system. MAML relies on meta-optimization through gradient descent in an architecture agnostic way. It expects to learn an initial representation that can be fine-tuned efficiently in a few steps.
- Prototypical Network (PN): a state-of-the-art metric-based meta-learning baseline. Given few samples, PN generates prototypes (feature representations) for each class and it uses the Euclidean distance metric to search the closest prototype as predicted class.
- RF-Net*: our proposed meta-learning framework without the RF metric module.
- RF-Net: our proposal. The comparison between RF-Net and RF-Net* highlights the impact of RF metric module.

4.3.2.3 Implementation

We first apply data normalization to all datasets. Afterward, we sample 80% environments of each dataset into the training dataset $\mathcal{D}$ and the remaining into the testing dataset $\hat{\mathcal{D}}$, and we perform 10-fold cross-validation for each experiment. We report the average accuracy evaluated on the testing dataset under 1-, 2-, 3-shot scenarios, i.e., $N^{\text{shot}} = \{1, 2, 3\}$. We use the regularization technique batch normalization [116] to avoid overfitting. The following experiments are all programmed with PyTorch [158] and run on NVIDIA TESLA V100 with 16GB memory.

4.3.3 Hyperparameter Searching

We first conduct hyperparameter searching of the dual-path base network, and thus we could employ the final hyperparameters in the following experiments for the fairness comparison. To be specific, we explore the following aspects: selection of CNN backbone, hidden dimension size, and activation function selection. This section explores the aspects of the dual-path base network on the Wi-Fi dataset with 100 environments.

Figure 4.5: CNN backbone selection.

4.3.3.1 Backbone Selection

Since our base network leverages CNN backbone f^{backbone} for extracting spatial features, the selection of CNN-backbone is essential for building a good base network. Therefore, we evaluate multiple state-of-the-art CNN backbones with various system complexity implanted into our base network. As depicts in Figure 4.5, these include two shallow networks: 5 layers CNN (cnn5) (with 3 convolutional layers and 2 fully connected layers) and 8 layers AlexNet (alexnet8) [82], two medium deep networks: 18 layers ResNet (resnet18) [157] and 19 layers VGG (vgg19) [159], as well as two deep networks: 44 layers ShuffleNet (shufflenet44) [160] and 152 layers ResNet (resnet152) [157]. We observe that our choice resnet18 achieves the best overall accuracy compared with other backbones, particularly for one-shot HAR. We further study the system overhead (complexity, inference time, and memory usage) of resnet. As the complexity of resnet18 takes up around 85% of the whole system, its overhead can be roughly regarded as the system overhead. According to a recent benchmark [161], resnet18, with low system complexity and memory usage, achieves a low inference time, leading to an excellent real-time performance. Moreover, compared with other lightweight systems such as shufflenet44, resnet18 achieves a comparable performance when deployed to commercial edge devices. Therefore, we select a pre-trained ResNet 18 layers network as the CNN backbone for RF-Net.

Figure 4.6: Hidden dimension size.

Figure 4.7: Activation function selection.

4.3.3.2 Hidden Dimension Size & Activation Function Selection

As described in Section 4.2.3, we use a hidden variable α to compose our network. Therefore, we aim to search a good dimension size. We separately explore three settings $\alpha = 128$, $\alpha = 256$, and $\alpha = 512$, and report the accuracy in Figure 4.6. Although $\alpha = 128$ is compared well with $\alpha = 512$ in average, $\alpha = 512$ outperforms $\alpha = 128$ on one-shot HAR. In this paper, one-shot HAR takes precedence, and thus we set α to 512. Afterwards, we also search the best activation function in our network. We explore ReLU, Leaky ReLU, Sigmoid, and Tanh. Figure 4.7 plots the performance. We observe that ReLU outperforms other function and we set all the activation function to ReLU. Therefore, we will employ pre-trained ResNet with 18 layers as our CNN backbone, set hidden dimension to 512 and select ReLU as activation function in the following experiments.

Figure 4.8: Overall comparison results for dual-path base network evaluation on Wi-Fi dataset.

4.3.4 Superiority of Proposed Network

This section will investigate the superiority of dual-path base networks and RF-Net over baselines on various RF sensing datasets. Specifically, we set the following parameters: M' to 12, batch size to 3, learning rate to 0.001, the number of epochs to 20, and we employ Adam optimizer to train networks.

4.3.4.1 Superiority of Dual-Path Base Network over Baselines

Figure 4.8, Figure 4.9, and Figure 4.10 separately report the accuracy of dual-path base network and the baselines on Wi-Fi, FMCW, and IR datasets. In all the cases, our base network outperforms all the baselines. It shows the effectiveness of both spatial and temporal features extracted by the dual-path base network for the purpose of HAR. To be specific, the performance of SPATIAL-FUSE outperforms TIME and FREQ remarkably. It validates our motivation that the utilization of RF signal matrix in dual-path (both time and frequency domain) is beneficial for HAR. In our experiments, we find that SPATIAL-FUSE is far better than SPATIAL-SEP when accuracy and time complexity are taken into account. One possible reason is that using the CNN backbone twice doubles the number of parameters in the network and makes SPATIAL-SEP relatively harder to train and converge.

In most cases, we observe that the performance of DUAL-wo-ATT is still unacceptable in spite of already involving temporal features. By contrast, the outstanding performance of dual-path base network demonstrates the efficacy of attended temporal features. We

Figure 4.9: Overall comparison results for dual-path base network evaluation on FMCW dataset.

Figure 4.10: Overall comparison results for dual-path base network evaluation on IR dataset.

further plot attention map $\mathcal{A}$ of two different activities on FMCW datasets in Figure 4.11. This showcases that the attention mechanism used in the temporal module is able to discover a rich joint temporal representation across two domains, thus improving the performance of HAR. In general, the accuracy of our dual-path base network outperforms DUAL-wo-ATT by around 4.2%.

Remark: One may intuitively expect a monotonic increase in accuracy given more shots. However, as the base network is designed for RF-HAR in better exploiting the input of high-dimensional RF signal matrices, its performance is optimized for general classification to fit into the overall meta-learning framework. Therefore, the fluctuation in accuracy across a few shots is reasonable and can be expected. We further demonstrate the efficacy of the dual-path base network for the general classification. As shown in

Figure 4.11: FMCW attention map $\mathcal{A}$ of two different activities: walking and sitting down.

Figure 4.12: Overall comparison results for dual-path base network evaluation on Wi-Fi, FMCW, IR datasets in terms of general classification.

Figure 4.12, the dual-path base network performs still outstanding consistently in the general classification on all RF datasets. Figure 4.12 (albeit not a fair comparison due to the separated datasets for different RF sensing techniques) also sheds light on the intrinsic differences among the three RF signals. Wi-Fi with narrowband cannot provide a high resolution in the time domain, but Doppler shift can still be retained in the frequency domain. Therefore, the overall performance of Wi-Fi is worse than the others, but the accuracy in the frequency domain is better than that in the time domain. Resorting to the power of wideband, FMCW and IR both achieve better performance due to a higher time resolution, yet FMCW slightly outperforms IR thanks to a higher tx power.

Figure 4.13: Overall comparison results for RF-Net evaluation on Wi-Fi dataset.

Figure 4.14: Overall comparison results for RF-Net evaluation on FMCW dataset.

Figure 4.15: Overall comparison results for RF-Net evaluation on IR dataset.

4.3.4.2 Superiority of RF-Net over Baselines

The comparison results for RF-Net on Wi-Fi, FMCW, and IR are listed in Figures 4.13, 4.14, and 4.15. We observe that our proposed RF-Net network has proved superiority over base-

Figure 4.16: Accuracy density plot of all evaluated networks.

lines on all three RF sensing techniques. In particular, compared RF-Net with RF-Net*, overall performance is enhanced by RF metric module and relatively steady. It highlights the efficacy of the RF metric module. This performance improvement gives credit to the trainable linear mapping of the RF metric module. As demonstrated in Section 4.2.2.1, its convex loss function should allow the RF metric module to learn an optimal metric for RF-Net. One may argue that all three RF techniques achieve a rather low accuracy, but this is the best one may obtain so far when adapting to new environments with very few labeled samples; in fact, our performance is already better than that for image recognition [162], possibly thanks to the depth information obtained by RF sensing.

In Figure 4.16, we further plot the accuracy density plot of RF-Net along with the dual-path base network and all the baselines. The best performance is highlighted in bold. The density plot depicts the accuracy distribution on all three RF datasets with $N^{\text{shot}} = \{1, 2, 3\}$. Intuitively, a dual-path base network notably outperforms its baselines. However, it is still far below the expectation. Compared with the dual-path base network, the density plot of RF-Net, along with its meta-learning baselines and fine-tuning baselines, exhibit some notable right skew. It indicates that RF-Net and its baselines, as expected, significantly improve the implanted base network. Nevertheless, RF-Net boosts the performance maximally among them: it has greatly helped the base network adapting to a new environment with even a single sample.

Figure 4.17: Performance on RF sensing types.

Figure 4.18: Performance heatmap: accuracy improvement from SPATIAL-SEP, while a darker color denotes a higher accuracy improvement.

4.3.4.3 Wi-Fi vs. IR vs. FMCW

To further evaluate the performance on each RF sensing dataset, we first plot the performance of the proposed networks and their baselines on Wi-Fi, IR, and FMCW in Figure 4.17. Meanwhile, we further generate a more representative performance heatmap in Figure 4.18 to help to visualize varying degrees of improved performance by RF-Net in each RF sensing dataset. From these two figures, we observe that RF-Net achieves the

Figure 4.19: Impact of number of environments.

highest overall performance in all datasets, and the performance of all concerned networks differs significantly across different RF sensing datasets. One curious sample on the evaluation over IR dataset is that all networks seem to achieve similar performances with and without meta-learning. The reason is that the intrinsic advantage of IR being not sensitive to environment impact (thus allowing the environment-free features to be captured via base network), yet RF-Net still performs the best among all networks. To summarize, based on the average accuracy shown in Figures 4.17 and 4.18, our proposed RF-Net is demonstrated to offer the most robust generalization that may fit all intrinsic data properties of all RF signals.

4.3.4.4 Impact of Number of Environments

We examine networks from the perspective of environmental diversity as well. The motivation is to explore how many environments could be handled for each network. We report the average accuracy of RF-Net and its baselines on the Wi-Fi dataset with 80, 100, and 120 environments in Figure 4.19. With the number of environments increasing, we observe that the performance of both fine-tuning and meta-learning baselines (MAML and PN) have dropped. Our RF-Net shows more steady trends. This performance indicates that RF-Net can deal with more environments without sacrificing performance severely, i.e., it is a more robust approach.

4.3.5 Limitations

Last but not least, we would like to discuss the limitation of RF-HAR. In this work, we validate the efficacy of RF-HAR on Wi-Fi, FMCW, and IR datasets. We set the number of activity categories N^{class} to 6. If more activity categories are required to be recognized, we doubt that the performance of our proposal may degrade. [163] evaluate the performance of well-known meta-learning algorithms on solving many-class few-shot classification. Based on their empirical findings, traditional meta-learning approaches can achieve promising results when $N^{\text{class}}=5$. However, when N^{class} is set to 50, their performance decreases severely. This degradation is attributed to the increasing difficulty of classification problems, especially in low data regimes. Therefore, current meta-learning methods often set N^{class} usually less than 10. If N^{class} is set to a large number, the meta framework of RF-HAR could be modified from the perspective of hierarchical classification to achieve adequate results. It can be used to train coarse-class and fine-class classifiers. Fine-class classifier applicants may be narrowed down using coarse-class classifiers. As a result, the difficulty of many-class few-shot classification can be alleviated.

4.4 Chapter Summary

In this chapter, we have utilized knowledge transfer via meta-learning for delivering the capability to RF-HAR system of being adaptive to a new environment with very few labeled data. We have proposed RF-Net, a unified metric-based meta-learning based neural network for one-shot RF-based HAR. It includes a dual-path HAR base network for learning influential features from general RF signals and a metric-based meta-learning framework for endowing adaptation capability with limited labeled data by transferring knowledge from seen environments. We have conducted hyperparameter searching to explore the best framework structures along with training settings. Moreover, we have performed extensive comparison experiments on three representative RF sensing techniques, Wi-Fi radio, FMCW radio, and IR, in multiple real-world indoor environments; compared with state-of-the-art baselines, all results have demonstrated the superiority of RF-Net being adaptive to a new environment in data scarcity.

Chapter 5

Robustness Knowledge Transfer via Meta-Learning against Both Data Scarcity and Adversarial Attacks¹

In this chapter, we are inspired by the property validated in Chapter 4 that knowledge transfer based learning systems can transfer knowledge from known tasks to new tasks and deliver the potential of fighting data scarcity. We employ this idea to enhance the robustness of IC system in the face of data scarcity and adversarial attacks. We regard “system robustness” as a type of knowledge and transfer it from a robust adversarially-trained (source) Image Classification (IC) system to a non-robust (target) IC system, while enhancing the robustness of the target IC system in the face of adversarial attacks. Consequently, we propose a meta-learning based adversarial defense framework named Meta Robustness Transfer (MRT) empowered by a non-parametric Robust Feature Selection Layer (RFSL) to perform a robustness transfer from adversarially-trained robust IC system while fighting data scarcity. We first provide an overview of related work. Then, we describe each component of MRT and RFSL. Afterward, through experiments in Mini-ImageNet, Caltech-UCSD Birds, and CIFAR-FS, we validate the efficacy of our approach on transferring robustness across tasks with different system architectures, different label spaces, and different datasets in extremely low data regimes.

¹The work in this chapter has been submitted to 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

5.1 Related Work

Inspired by knowledge transfer that learns knowledge from known tasks and adapts it to new tasks, few studies employ this idea to enhance IC system robustness in defending against adversarial examples by regarding “system robustness” as a type of knowledge. The first research direction is adversarial transfer learning. Related studies all aim to directly acquire robustness from a robust adversarially-trained (source) system to a non-robust (target) system [38, 39]. Consequently, they can avoid re-conduct adversarial training for the non-robust system and circumvent the requirement of crafting adversarial examples. The second research direction is adversarial meta-learning [164, 165]. They perform meta-learning based knowledge transfer in the face of data scarcity as described in Sec. 4.1.2, while crafting adversarial examples in meta-training phase. Essentially, these studies are the variants of adversarial training, and they endow the capability of fighting data scarcity.

5.1.1 Adversarial Transfer Learning

Shafahi *et al.* [38] employ model-based transfer learning. They make the following observations: i) if the source system used for transfer learning is robust, the target system is also robust but less than the source, and ii) robustness transfers between systems that are more similar like similar data distributions, robustness transfers better. Therefore, they intend to use a robust feature extractor from the source system as a basis for the non-robust system. Afterward, they add untrained layers on top of the basis and fine-tune these untrained layers with target labeled data samples. Essentially, they employ pre-trained system weights as the medium of transfer. Their approach is architecture-specific, but they can transfer robustness across the different datasets.

Chan *et al.* [39] design Input Gradient Adversarial Matching (IGAM) to transfer robustness by employing input gradient map as a medium. Compared with [38], they are not constrained by the assumption that the source system and target system have similar architecture as pre-trained system weights are the medium of transfer. IGAM is inspired by the fact that robust adversarially-trained systems show more interpretable input gradients while standard trained systems display noisy input gradients. Essentially,

the value of the input gradient at each pixel represents how a small change there can affect the system’s output. From this perspective, it can be roughly regarded as to how vital each pixel is for prediction. IGAM intends to mimic how robust systems view importance on images through input gradients, resulting in robust systems.

Specifically, a robust adversarially-trained system r_ϕ parameterized by ϕ is first provided. When it is adapted to target system, r_ϕ is fine-tuned with labeled target-task data samples first. Afterward, input gradient map of fine-tuned r_ϕ is computed as follows:

$$\mathcal{J}_\phi = \nabla_x \mathcal{L}_\phi^{\text{CE}}, \quad (5.1)$$

where $\mathcal{J}_\phi$ denotes input gradient map of r_ϕ and $\mathcal{L}_\phi^{\text{CE}}$ represents cross entropy loss. Similarly, $\mathcal{J}_\theta$ of target system f_θ is calculated as well, where $\mathcal{J}_\theta = \nabla_x \mathcal{L}_\theta^{\text{CE}}$.

Afterward, f_θ is trained with an adversarial objective, inspired by GAN. To be specific, f_θ is trained to compute $\mathcal{J}_\theta$, which is hard to be distinguished from $\mathcal{J}_\phi$ by a discriminator $f^{\text{discriminator}}$. Moreover, a l_2 reconstruction regularization is involved to further penalize the difference between $\mathcal{J}_\theta$ and $\mathcal{J}_\phi$. Subsequently, f_θ is motivated to compute a similar input gradient map as the robust r_ϕ , leading to robustness transfer. Consequently, IGAM can be employed to transfer robustness across different architectures and different datasets.

5.1.1.1 Drawbacks

The approaches above have the following limitations. [38] employ parameter weights as a medium to transfer robustness across different architectures and different datasets. It poses a limitation on architectures of the target system, whose shallow architecture is required to be consistent with the one of the source system. Although [38,39] can improve the robustness of IC system by avoiding adversarial examples crafting, they fail to take into account low data regimes where the transfer performance could decay severely [38].

5.1.2 Adversarial Meta-Learning

ADversarial Meta-Learner (ADML) [165] is actually an adversarially-trained MAML [143]. It crafts adversarial examples $\mathcal{D}^{\text{adv}}$ from $\mathcal{D}$, and then uses them to augment the datasets for improving the robustness. Specifically, they use FGSM to craft adversarial examples.

They state that ADML can utilize correlations between clean samples and adversarial examples to motivate the gradient update and obtain a good initialization of f_θ , which defend against adversarial attacks.

Formally, in meta-training phase, for each $\mathcal{T}_i \sim P(\mathcal{T})$, ADML randomly split $\mathcal{D}^{\text{adv}}$ into $\{\mathcal{D}^{\text{adv,S}}, \mathcal{D}^{\text{adv,Q}}\}$ and $\mathcal{D}$ into $\{\mathcal{D}^{\text{S}}, \mathcal{D}^{\text{Q}}\}$. Afterward, they separately adapt parameter with gradient descent on clean samples and adversarial examples as follows:

$$\theta_i^{\text{clean}} \leftarrow \theta - \nu^{\text{clean,task}} \nabla_{\theta} \mathcal{L}^{\mathcal{T}_i}(f_{\theta}; \mathcal{D}^{\text{S}}), \quad (5.2)$$

$$\theta_i^{\text{adv}} \leftarrow \theta - \nu^{\text{adv,task}} \nabla_{\theta} \mathcal{L}^{\mathcal{T}_i}(f_{\theta}; \mathcal{D}^{\text{adv,S}}), \quad (5.3)$$

where $\nu^{\text{clean,task}}$ and $\nu^{\text{adv,task}}$ denote the step size of task objective optimization on clean samples and adversarial examples, respectively.

Consequently, meta objective optimization is conducted as follows:

$$\theta \leftarrow \theta - \nu^{\text{clean,meta}} \nabla_{\theta} \sum_{\mathcal{T}_i \sim P(\mathcal{T}_i)} \mathcal{L}^{\mathcal{T}_i}(f_{\theta_i^{\text{clean}}}; \mathcal{D}^{\text{Q}}), \quad (5.4)$$

$$\theta \leftarrow \theta - \nu^{\text{adv,meta}} \nabla_{\theta} \sum_{\mathcal{T}_i \sim P(\mathcal{T}_i)} \mathcal{L}^{\mathcal{T}_i}(f_{\theta_i^{\text{adv}}}; \mathcal{D}^{\text{adv,Q}}), \quad (5.5)$$

where $\nu^{\text{clean,meta}}$ and $\nu^{\text{adv,meta}}$ mean the step size of meta objective optimization on clean samples and adversarial examples, respectively.

Adversarial Querying (AQ) [164] proposes a more efficient approach that only perturbs $\mathcal{D}^{\text{Q}}$ for data augmentation. Specifically, for each $\mathcal{T}_i \sim P(\mathcal{T})$, AQ first splits $\mathcal{D}$ into $\{\mathcal{D}^{\text{S}}, \mathcal{D}^{\text{Q}}\}$, and then system f_θ is fine tuned on $\mathcal{D}$. Afterward, they craft adversarial query set $\mathcal{D}^{\text{adv,Q}}$ from clean query set $\mathcal{D}^{\text{Q}}$ by maximizing $\mathcal{L}_\theta^{\text{CE}}$ and constraining $\|\mathbf{x}^{\text{adv,Q}} - \mathbf{x}^{\text{Q}}\|_p < \epsilon$. Gradient with respect to parameter $\mathcal{G}_i$ is conducted as follows:

$$\mathcal{G}_i = \nabla_{\theta} \mathcal{L}^{\mathcal{T}_i}(f_{\theta}; \mathcal{D}^{\text{adv,Q}}). \quad (5.6)$$

Subsequently, meta objective optimization is conducted as follows:

$$\theta \leftarrow \theta - \nu^{\text{meta}} \sum_{\mathcal{T}_i \sim P(\mathcal{T}_i)} \mathcal{G}_i. \quad (5.7)$$

5.1.2.1 Drawbacks

Although meta-learning approaches [164, 165] achieve outstanding performance in data scarcity, they mainly investigate the robustness enhancement across tasks with different label spaces. Moreover, they are the variants of adversarial training rather than direct acquiring robustness from the adversarially-trained system. Consequently, they still suffer from crafting adversarial examples.

Figure 5.1: c-Score (y-axis) between each feature (x-axis) and the corresponding input gradient map computed on MNIST; the selected samples are all correctly classified. Note the remarkable similarity between (a) and (b) under an adversarially-trained system, which simply does not exist between (c) and (d) under a normally-trained system.

5.1.3 Our Proposal

To bridge the gap, our proposal employ knowledge transfer to transfer robustness in the form of robust features, from an adversarially-trained robust IC system to a non-robust IC system, in consideration of data scarcity. Moreover, it is expected to transfer robustness across different IC system architectures, different label spaces, and different

datasets. Two critical observations have largely inspired our design: i) the robustness of adversarially-trained system stems from learning *robust features* [44] and ii) compared with the normally-trained system, the adversarially-trained system leads to more interpretable input gradient maps [166]. Therefore, we define an attention-based *correlation score* (c-Score) to characterize the relationship between these two observations on an adversarially-trained system. As shown in Figure 5.1, the adversarially-trained system shows a much higher similarity in the c-Score distribution for both clean and adversarial examples, proving the relevance of c-Score distribution to the robustness of a system.

Consequently, we regard the c-Score distribution as the medium to transfer robustness, and we propose a non-parametric Robust Feature Selection Layer (RFSL) to learn this distribution (especially its cross-sample similarity) from a robust system. In order to handle data scarcity, we further propose Meta Robustness Transfer (MRT) as a meta-learning based adversarial defense framework to achieve a few-shot robustness transfer. We adapt the metric-based approach to our MRT: it learns meta knowledge as a function to embed samples into robust features so that employing meta knowledge in a target system reduces sample classification to simple similarity comparison, allowing robustness to be transferred with very few labeled data. We select metric-based meta-learning as it is much more efficient than others, as explained in Sec. 4.1.3.

5.2 Meta Robustness Transfer (MRT)

In this section, we first provide an overview of MRT, then we introduce RFSL. Finally, we elaborate our training strategy.

5.2.1 Overview

Figure 5.2 illustrates our MRT framework with a two-phase meta-training strategy. Essentially, given a robust adversarially-trained IC system, MRT trains a non-robust system for transferring robustness from the robust system. The whole training process is divided into two phases: i) *MRT phase I* leverages the robust system to guide RFSL in learning the way of robust feature embedding, and ii) *MRT phase II* meta-trains the whole adapted system to embed input images into robust features that are further utilized to classify samples via a similarity comparison.

Figure 5.2: Overview of MRT framework: it transfers robustness from robust (source) system to data-scarce non-robust (target) system, driven by a two-phase meta-training strategy.

5.2.2 Robust Feature Selection Layer

We propose RFSL as an attention-based non-parameteric module to learn the *correlation score* (c-Score) between two characteristics of adversarially-trained systems, namely robust features and more interpretable input gradient maps. Figure 5.3 shows the generic form of RFSL. Given input map $M \in \mathbb{R}^{c^{\text{img}} \times h^{\text{img}} \times w^{\text{img}}}$ and c -channel features $H \in \mathbb{R}^{c \times h \times w}$, RFSL aims to compute the c-Score $\eta \in \mathbb{R}^c$. To obtain a multi-perspective correlation, RFSL first derives correlations separately from three perspectives: channel-wise, height-wise, and width-wise, and then it aggregates the results to get η .

Take the channel-wise perspective as an example: input map M and features H are first reshaped and concatenated into *key* $K \in \mathbb{R}^{c^{\text{img}} \times h^{\text{img}} \times w^{\text{img}} \times hw}$ and *observation* $O \in \mathbb{R}^{c^{\text{img}} \times c \times hw}$. Then an attention score $\eta^{\text{channel}} \in \mathbb{R}^{c^{\text{img}} \times c \times h^{\text{img}} \times w^{\text{img}}}$ is calculated via c^{img} channel matrix multiplication. Finally, height and width information are compressed via a mean operation, leading to $\eta^{\text{channel}} \in \mathbb{R}^{c^{\text{img}} \times c}$ for the channel-wise perspective. Meanwhile, height-wise attention score $\eta^{\text{height}} \in \mathbb{R}^{h^{\text{img}} \times c}$ and width-wise attention score $\eta^{\text{width}} \in \mathbb{R}^{w^{\text{img}} \times c}$ are generated in the same vein. In the end, η^{channel} , η^{height} , and η^{width} are concatenated into $\eta \in \mathbb{R}^{d \times c}$ where $d = c^{\text{img}} + h^{\text{img}} + w^{\text{img}}$, then η is scaled into the range $[0, 1]$ as follows:

$$\eta = \sigma_x \left(\left\| z \left(\frac{1}{d} \sum_{i=1}^d \eta_{ij} \right) \right\| \right), \quad (5.8)$$

Figure 5.3: Overview of RFSL.

where z refers to standardization and σ_x denotes the Softmax function. The reason of using scaling is to stabilize the learning process. More details will be provided in Sec. 5.2.3.

5.2.3 Adapted System via Meta-Learning

As shown in Figure 5.2, with the help of RFSL, MRT aims to transfer robustness from a robust system r_ϕ parameterized by ϕ to a non-robust system f_θ parameterized by θ , and it is trained with a two-phase meta-training in each iteration. The training iterations are performed on the specially designed task set $\mathcal{T} = \{(\mathcal{D}^S, \mathcal{D}^Q)_i\}_{i=1, \dots, Z}$.

5.2.3.1 MRT Training Phase I

In the first phase, given $(\mathbf{x}, y) \sim \mathcal{D}^S$, the robust system r_ϕ are first employed to compute an input gradient map through back propagation as $\nabla_x \mathcal{L}_\phi$, where $\mathcal{L}_\phi$ stands for cross entropy loss and dimension of $\nabla_x \mathcal{L}_\phi$ is same as $\mathbf{x} \in \mathbb{R}^{c^{\text{img}} \times h^{\text{img}} \times w^{\text{img}}}$. Meanwhile, the non-

robust embedding network $f_\theta(\mathbf{x}) : \mathbf{x} \mapsto \mathbb{R}^{c \times h \times w}$ that maps input $\mathbf{x}$ to features H_0 is also employed.

Afterwards, RFSL is utilized to separately compute c-Scores η^r and η^f from r_ϕ and f_θ , respectively. To be specific, η^r via $\text{RFSL}(\nabla_x \mathcal{L}_\phi, H_0)$ and η^f via $\text{RFSL}(\mathbf{x}, H_0)$ are computed. H_0 is selected to calculate η^r for architecture agnostic purposes. Moreover, $\mathbf{x}$ is directly employed to generate η^f instead of using the input gradient map because the map computed by f_θ cannot be interpretable initially, causing trouble in bootstrapping the training. Although input gradient map would gradually become interpretable during training, we still stick to $\mathbf{x}$ through the whole process to maintain the end-to-end training style.

C-Score Losses To exploit this score, we employ Mean Squared Error (MSE) loss $\mathcal{L}^{\text{score}}$ to minimize the ℓ_2 distance between c-Score η^r and η^f as follows:

$$\min \mathcal{L}^{\text{score}} = \min \|\eta^r - \eta^f\|_2^2. \quad (5.9)$$

Additionally, we also utilize a discriminator h_α parameterized by α to discriminate source η^r and target η^f with a cross entropy loss $\mathcal{L}^{\text{adv}}$ during training. f_θ is optimized in order to minimize $\mathcal{L}^{\text{score}}$ and to maximize $\mathcal{L}^{\text{adv}}$. It enforces η^f to be indistinguishable from η^r in distribution sense.

Robust Feature Selection We assume that c-Score hints a way of robust feature selection as well. Motivated by this hypothesis, η^f is utilized to weight H_0 to compute weighted feature H . We expect that weighted H shall still maintain the classification performance. Moreover, after scaling, it shall have similar distribution shape as η^f . To achieve these objectives, we first apply Kullback–Leibler (KL) divergence $\mathcal{L}^{\text{dist}}$ to assess the distribution distance between scaled c-Score $\eta^{f, \text{scaled}} = \sigma_x(\eta^f)$ and scaled weighted features $H^{\text{scaled}} = \sigma_x\left(\frac{1}{hw} \sum_{k=1}^w \sum_{j=1}^h H_{ijk}\right)$. We aim to minimize $\mathcal{L}^{\text{dist}}$ as follows:

$$\min \mathcal{L}^{\text{dist}} = \min \sum_{i=1}^c H_i^{\text{scaled}} \log \frac{H_i^{\text{scaled}}}{\eta_i^f}. \quad (5.10)$$

Afterwards, we insert a classification layer on top of f_θ and minimize cross entropy loss $\mathcal{L}_\theta$ to enhance the classification performance of H . In summary, the final objective of adapted system in MRT Phase I is:

$$\begin{aligned}\theta^* &= \operatorname{argmin}_{\theta} \mathcal{L}_\theta + \lambda^{\text{score}} \mathcal{L}^{\text{score}} + \lambda^{\text{dist}} \mathcal{L}^{\text{dist}} - \lambda^{\text{adv}} \mathcal{L}^{\text{adv}}, \\ \alpha^* &= \operatorname{argmin}_{\alpha} \mathcal{L}^{\text{adv}},\end{aligned}\tag{5.11}$$

where λ^{score} , λ^{dist} , and λ^{adv} are regularization parameters.

5.2.3.2 MRT Training Phase II

After phase I, the optimal embedding network f_θ with RFSL as modified embedding network $g_\theta(x) : \mathbf{x} \mapsto \mathbb{R}^{c \times h \times w}$ is passed to MRT Phase II. The modified embedding network now becomes the following:

$$g_\theta(\mathbf{x}) = \text{RFSL}(\mathbf{x}, f_\theta(\mathbf{x})) * f_\theta(\mathbf{x}).\tag{5.12}$$

In this phase, framework intends to learn the way of classifying query sample via simple similarity comparison. To be specific, $\mathbf{x}$ could be predicted by a weighted sum of the labels in $\mathcal{D}^S$ as follows:

$$P(y|\mathbf{x}, \mathcal{D}^S) = \sum_{(\mathbf{x}_i, y_i) \sim \mathcal{D}^S} \mathcal{M}(g_\theta(\mathbf{x}), g_\theta(\mathbf{x}_i)) y_i,\tag{5.13}$$

where $\mathcal{M}$ is metric function for measuring the similarity between robust features $g_\theta(\mathbf{x})$ and $g_\theta(\mathbf{x}_i)$. Therefore, the training objective function of phase II can be formulated as follows:

$$\theta^* = \operatorname{argmax}_{\theta} \mathbb{E} \left[\sum_{(\mathbf{x}, y) \sim \mathcal{D}^Q} P(y|\mathbf{x}, \mathcal{D}^S) \right].\tag{5.14}$$

Remark: After two-phase MRT-training, the optimal adapted system is finally evaluated on unseen tasks; the procedure is same as Eqn. (5.13).

5.3 Experiments

In this section, we validate that our proposed MRT can transfer robustness across tasks with different system architectures, different label spaces, and different datasets under few-shot settings.

As few-shot learning aims to obtain good performance given limited supervised information, its performance is below general classification performance, so we target only at a better performance over existing proposals under the same setting. Moreover, we would like to emphasize that our objective is to demonstrate the ability of robustness transfer rather than to search for the best structure as the embedding function f_θ . We employ the metric function proposed in prototypical network [147] to implement Eqn. (5.13). It calculates $P(y|\mathbf{x}, \mathcal{D}^S)$ based on a Softmax over distances to the *prototypes*, i.e., mean vector of $\mathcal{D}^S$ in the embedding space. Note that MRT is compatible with other well-known metric-based meta-learning frameworks.

We mainly compare MRT with Adversarial Querying (AQ) [164] enhanced system and normally-trained (NT) system. The reason to select AQ as the main comparison target is that its performance has been demonstrated better than ADversarial Meta-Learner (ADML) [148] and adversarial transfer learning [38]. Compared with MRT, both AQ and NT perform the same meta-training shown by Eqn. (5.13) and (5.14), except that AQ additionally perturbs query samples to augment the datasets for improving the robustness. For training AQ, we employ 7-step l_∞ Projected Gradient Descent (PGD) attack with $\epsilon = 0.031$ and step = 0.0078. We also evaluate Input Gradient Adversarial Matching (IGAM) [39] under the same settings, but it fails to achieve comparable performance, so we only provide the related evaluation results in Appendix B.1, instead of listing them in this section.

We evaluate the transferred robustness in the face of four white-box attacks: 7-step l_∞ PGD-7 ($\epsilon = 0.031$ and step = 0.0078), 40-step l_∞ PGD-40 ($\epsilon = 0.031$ and step = 0.0078), l_∞ Fast Gradient Sign Method (FGSM) ($\epsilon = 0.031$), and L_∞ Carlini & Wagner (C&W) ($\epsilon = 0.031$). Through hyperparameter searching, we set $\lambda^{\text{score}} = 1e+4$, $\lambda^{\text{dist}} = 1$, $\lambda^{\text{adv}} = 1$. We utilize Adam optimizer [151] for training and set its learning rate to 1e-3.

Table 5.1: Accuracy (%) of ablation studies.

Loss	Clean	PGD-7	RFSL	Clean	PGD-7
L_θ	61.64	14.75	Channel	55.98	18.30
$+L^{\text{dist}}, L^{\text{score}}$	-7.77	+7.21	+Height	+3.39	-5.00
$+L^{\text{adv}}$	+2.50	-1.05	+Width	-3.00	+7.61

5.3.1 Ablation Studies

We conduct ablation studies to validate the efficacy of the proposed loss functions and multi-perspective RFSL module. We evaluate their efficacy on robustness transfer across datasets and classification tasks. We select Resnet18 [3] adversarially-trained on Mini-ImageNet [145] as robust source model with 7-step L_∞ PGD ($\epsilon = 0.031$ and step = 0.0078) like [34]. Mini-ImageNet is a dataset with a total of 60,000 images (80 training classes and 20 testing classes), each of size $84 \times 84 \times 3$. We take Resnet18 (except last classification layer) as the target model f_θ . CIFAR-FS [148] is selected to build the target dataset (64 training classes and 20 testing classes), each of size $32 \times 32 \times 3$. We sample 3-way 5-shot tasks from the 64 classes and the 20 classes to separately build target training dataset and target testing dataset.

Table 5.1 demonstrates the efficacy of our proposed loss functions and multi-perspective RFSL module. Firstly, after involving L^{dist} and L^{score} , robustness (performance on adversarial samples) is increased by 7.21%, while clean accuracy is decreased. Further involving L^{adv} improves the trade-off between performance on clean and adversarial samples. We also observe that RFSL with channel, height, and width perspectives allow a fine-tuning of performance trade-off between robustness and clean accuracy.

5.3.2 Transfer across Different Architectures

Experiment Setting Here MRT aims to transfer robustness across different system architectures and label spaces (classification tasks) on Mini-ImageNet [145]. The robust system used in this section is an adversarially-trained Resnet18 [3] with 7-step L_∞ PGD attack ($\epsilon = 0.031$ and step = 0.0078) like [34]. Robust system achieves 59.0573% and 26.8099% accuracy on classifying clean samples and adversarial examples respectively. For the non-robust system, we consider Mobilenet [167] and Resnet34 [3] (both except

Table 5.2: Accuracy (%) of Mobilenet as f_θ under 3-way 1-shot tasks on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	40.120	0.160	0.040	22.840	0.157
AQ	36.310	8.010	0.750	29.280	0.000
MRT	37.540	16.670	8.370	28.350	36.690

Table 5.3: Accuracy (%) of Mobilenet as f_θ under 3-way 5-shot tasks on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	57.600	0.000	0.000	14.300	0.000
AQ	44.120	6.080	3.398	31.30	0.420
MRT	44.910	11.580	2.275	19.950	19.960

last classification layer) as f_θ to investigate the robustness transfer performance from two perspectives. We regard the transfer to Mobilenet with 235K parameters as *downward transfer* since its system complexity is 5 times lower than the robust Resnet18 (1122K). Conversely, transferring robustness from Resnet18 (1122K) to Resnet34 (2133K) is treated as *upward transfer*.

Results For both downward and upward transfers, we have the following observations: i) Comparing the performance of 5-shot tasks (Tables 5.3 and 5.5) with 1-shot tasks (Tables 5.2 and 5.4), we notice that more support samples have a positive impact on classifying clean query samples for all three methods, but this may not apply to adversarial examples. ii) NT achieves the best performance on clean samples in all tasks, but it fails to defend against adversarial attacks. iii) Although both AQ and MRT have inevitable trade-offs between generalization (performance on clean samples) and robustness (performance on adversarial examples), MRT strikes a better balance between them. Finally, iv) compared with AQ, MRT shows a more steady performance on adversarial examples crafted by different attacks.

In particular, Tables 5.2 and 5.3 showcase the downward transfer performance. In 1-shot tasks, MRT outperforms AQ on classifying both clean and adversarial examples crafted by powerful PGD and C&W attacks; MRT can achieve the almost same performance in defending against 1 step FGSM attack. As for 5-shot tasks, though AQ shows a significant improvement in defending against FGSM attack, its overall performance on

Table 5.4: Accuracy (%) of Resnet34 as f_θ under 3-way 1-shot tasks on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	49.650	0.010	0.000	0.320	0.027
AQ	36.310	3.255	2.130	7.712	36.060
MRT	36.880	14.820	12.780	22.520	34.010

Table 5.5: Accuracy (%) of Resnet34 as f_θ under 3-way 5-shot tasks on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	62.690	0.000	0.000	0.000	0.000
AQ	41.480	14.260	13.640	21.580	35.110
MRT	45.970	12.470	7.250	21.670	36.250

classifying clean samples and other adversarial examples is still lower than our MRT. In the upward transfer, similar findings can be drawn from Tables 5.4 and 5.5: MRT outperforms AQ on both generalization and robustness performance in 1-shot tasks, though AQ shows more improvement in 5-shot tasks. Comparing upward transfer with downward transfer, we notice that AQ’s performance highly relies on data volume and the selection of embedding networks. In contrast, MRT presents a more steady performance.

We further evaluate model robustness under PGD-7 attacks with different perturbation budgets ϵ from 4/255.0 to 16/255.0. As shown in Table 5.6, MRT consistently outperforms AQ on all perturbation budgets. Moreover, we observe that the accuracy of AQ decays significantly when the adversarial perturbation budget is increasing. Compared with AQ, MRT shows more robust to large attack budgets.

We also compare the computational time of these three approaches (especially for upward transfer). As shown in Figure 5.4a, a lower computational time is required to train MRT in all tasks. On average, MRT saves around 17 minutes. Apparently, an

Table 5.6: Accuracy (%) of defending against PGD-7 attacks with different perturbation budgets under 3-way 1-shot on Mini-ImageNet.

ϵ	4/255.0	8/255.0	10/255.0	16/255.0
MRT (Mobilenet)	19.6	16.67	13.1	12.05
MRT (Resnet34)	14.88	14.82	14	14
AQ (Mobilenet)	11.47	8.01	5.20	4.62
AQ (Resnet34)	7.567	3.255	1.923	1.433

(a) Training time.

(b) Total attacking time of performing all four attacks.

Figure 5.4: Time complexity comparison, given Resnet34 as f_θ on Mini-ImageNet.

attack may try to craft new adversarial examples based on the newly trained f_θ , so we also evaluate the *attacking time* spent on MRT compared with that on both AQ and NT. According to Figure 5.4b, the attacking time on MRT is 2 to 3 times that on AQ. As this quantity represents the challenge posed by a robustness transfer method to attackers searching for new adversarial examples, the results again indicate the strength and effectiveness of MRT. A longer attacking time may deter attackers from further attacking. Further experiments result of 5-way tasks are postponed to Appendix B.2.

5.3.3 Transfer across Different Datasets

Experiment Setting Here we evaluate MRT’s performance on transferring robustness across different datasets and label spaces (classification tasks). We consider downward transfer, same image size transfer, and upward transfer as well. We take Resnet18 (except last classification layer) as the non-robust system f_θ . For downward transfer, MRT transfers robustness from robust Resnet18 in Sec. 5.3.2 to Resnet18 on CIFAR-100 [148], a dataset specifically designed for few-shot setting by randomly sampling from

CIFAR-100 [168] that contains images of size $32 \times 32 \times 3$. As the input dimensions of Mini-ImageNet and CIFAR-FS are different, samples are resized into $84 \times 84 \times 3$ when robust system computes input gradient maps. Tasks are sampled from 64 classes for training and 20 classes for testing.

Table 5.7: Accuracy (%) of Resnet18 as f_θ under 3-way 1-shot on CIFAR-FS.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	42.340	11.360	9.869	19.080	12.730
AQ	36.270	21.060	20.310	29.660	19.240
MRT	37.840	20.070	18.940	24.180	35.160

For same image size transfer, MRT transfers robustness from robust Resnet18 used in Sec. 5.3.2 to Resnet18 on Caltech-UCSD Birds (CUB-200) [169] that involves 200 bird species. We resize each image of CUB-200 into $84 \times 84 \times 3$ to match that of Mini-ImageNet, relying on the Lanczos algorithm for resizing [170]. Tasks are sampled from 140 classes for training and the remaining 60 classes for testing. As for upward transfer, we first adversarially train robust Resnet18 on CIFAR-FS with 7-step l_∞ PGD ($\epsilon = 0.031$ and step = 0.0078). Robust system achieves 65.6328% and 46.1635% accuracy on classifying clean samples and adversarial examples, respectively. Afterwards, MRT transfers robustness from it to Resnet18 on Mini-ImageNet. Similarly, samples are resized into $32 \times 32 \times 3$ via the Lanczos algorithm when a robust system computes input gradient maps.

Results The observations summarized in Sec. 5.3.2 are displayed in this section as well. In particular, Tables 5.7 and 5.8 showcase the downward transfer performance. In 1-shot tasks, MRT is far better than AQ on classifying adversarial examples crafted by C&W; MRT can achieve similar performance on classifying clean samples and adversarial examples crafted by gradient-based adversarial attacks. As for 5-shot tasks, AQ sacrifices accuracy on clean query samples in return for robustness. In particular, compared with NT, AQ reduces clean accuracy by 24.88%, and its accuracy on classifying adversarial examples is up to 13.30 times that of NT’s. By contrast, MRT strikes a better balance by reducing the accuracy on classifying clean query samples by only 8.11% but achieving accuracy on classifying adversarial examples up to 16.24 times of NT’s.

Table 5.8: Accuracy (%) of Resnet18 as f_θ under 3-way 5-shot on CIFAR-FS.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	64.480	2.889	2.151	10.420	2.694
AQ	39.600	28.940	28.612	30.740	27.390
MRT	56.370	20.910	18.870	30.540	43.760

Table 5.9: Accuracy (%) of Resnet18 as f_θ under 3-way 1-shot on CUB-200.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	58.300	0.000	0.000	0.000	0.000
AQ	38.470	1.991	0.991	10.300	35.620
MRT	41.650	10.560	8.740	21.540	35.690

Tables 5.9 and 5.10 report the performance of same image size transfer. In 1-shot tasks, MRT outperforms AQ on classifying both clean query samples and adversarial examples crafted by all attacks. In 5-shot tasks, AQ performs similarly to those shown in downward transfer. Although AQ improves robustness, its classification performance on clean query samples is reduced by 28.86%. Tables 5.11 and 5.12 present upward transfer performance. In 1-shot tasks, MRT outperforms AQ in defending gradient-based attacks, and it obtains comparable results when attacked by C&W. In 5-shot tasks, AQ’s performance on gradient-based attacks is enhanced, but its average performance is still lower than MRT. Time complexity comparisons are postponed to Appendix B.3. We also postpone experimental results for 5-way tasks to Appendix B.2.

5.3.4 Black-Box Adversarial Robustness

In black-box attacks, instead of relying on gradients of the network, a corresponding substitute of the network is directly attacked by PGD-7 to generate the adversarial samples. We select MRT (Mobilenet), MRT (Resnet34), AQ (Mobilenet), and AQ (Resnet34) as substitutes. Then, MRT (Mobilenet), MRT (Resnet34), AQ (Mobilenet), and AQ (Resnet34) are test on the generated examples. We mainly evaluate black-box adversarial robustness on Mini-ImageNet. As demonstrated in Table 5.13 and Table 5.14, MRT (Mobilenet) achieves the best accuracy among these four networks.

Table 5.10: Accuracy (%) of Resnet18 as f_θ under 3-way 5-shot on CUB-200.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	67.860	0.000	0.000	0.000	0.000
AQ	39.000	14.760	13.210	21.590	32.900
MRT	55.060	9.371	7.250	21.890	43.520

Table 5.11: Accuracy (%) of Resnet18 as f_θ under 3-way 1-shot on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	54.010	0.000	0.000	0.000	0.000
AQ	38.520	3.019	1.936	8.454	37.720
MRT	36.540	12.540	10.590	20.970	35.110

Figure 5.5: Overall performance comparison. MI, CUB, and FS refer to Mini-ImageNet, CUB-200, and CIFAR-FS.

5.3.5 Overall Performance

The above results demonstrate that MRT can transfer robustness across different system architectures, different datasets, and different label spaces in extremely low data regimes. Compared with AQ, MRT requires a lower training time, but it achieves better performance in both generalization and robustness. We further plot Figure 5.5 to depict overall performance comparison in all tasks. This figure shows the accuracy of classi-

Table 5.12: Accuracy (%) of Resnet18 as f_θ under 3-way 5-shot on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	64.030	0.000	0.000	0.000	0.000
AQ	47.930	10.330	3.690	17.910	24.040
MRT	50.130	8.670	5.630	17.600	34.760

Table 5.13: Accuracy (%) of defending against black-box attacks under 3-way 1-shot on Mini-ImageNet.

Substitute to Attack	MRT (Mobilenet)	MRT (Resnet34)	AQ (Mobilenet)	AQ (Resnet34)
MRT (Mobilenet)	X	36.82	34.75	35.88
MRT (Resnet34)	36.7	X	34.98	35.12
AQ (Mobilenet)	36.15	36.88	X	34.62
AQ (Resnet34)	36.57	35.7	35.57	X
Average	36.47	36.47	35.1	35.21

fying clean samples and average accuracy of classifying adversarial examples crafted by PGD-7, PGD-40, FGSM, and C&W attacks. We observe that NT achieves the best performance on classifying clean samples in all tasks while being comprehensively defeated by adversarial examples. In the meantime, MRT outperforms AQ in all tasks.

Another interesting finding is that both AQ and MRT suffer from the trade-off between generalization and robustness: they improve robustness at the cost of reducing accuracy on classifying clean samples to various degrees [34], but AQ performs worse in the overall accuracy. The reason is that adversarial examples augmentation during training would reduce generalization performance inevitably, as generalization and robustness are fundamentally at conflict [166]. Fortunately, MRT strikes a better balance between generalization and robustness because it transfers robustness from an adversarially-trained robust system that, albeit being still under the inevitable trade-off, can be better trained given the sufficient data.

Specifically, the fact that MRT suffers less of the generalization-robustness trade-off stems from three causes. First, the robust system is not trained under the limitation of low data regimes. Second, MRT employs only clean samples throughout its training process, so the generalization capability of MRT shall be better than AQ. Last but not least, given a clean samples and corresponding adversarial examples, the embedding network with RFSL can map both inputs into more predictive and robust features with

Table 5.14: Accuracy (%) of defending against black-box attacks under 3-way 5-shot on Mini-ImageNet.

Substitute to Attack	MRT (Mobilenet)	MRT (Resnet34)	AQ (Mobilenet)	AQ (Resnet34)
MRT (Mobilenet)	X	43.91	44.48	37.35
MRT (Resnet34)	42.24	X	42.95	37.48
AQ (Mobilenet)	45.25	45.53	X	42.88
AQ (Resnet34)	45.59	43.2	36.88	X
Average	44.36	44.21	41.43	39.24

Figure 5.6: Qualitative results of Resnet34 as f_θ on Mini-ImageNet.

similar distributions. Because these MRT-learned features are used to classify both clean and adversarial examples in unseen tasks by a simple similarity comparison with samples from the support set, they largely maintain the nice property pertaining to the robust system.

As shown in Figure 5.6, given the support sample of each class, features generated from g_θ are very predictive. It shows that MRT can extract distinct features in the face of various samples. Moreover, given a clean query sample and corresponding adversarial sample, we observe that feature distributions are similar. The indistinguishable distributions between clean and adversarial examples benefit from the efficacy of MRT on transferring robustness. Furthermore, MRT also preserves the classification accuracy by allowing both the clean and adversarial examples in the query set to be correctly classified, which is visually clear from the feature distributions. All these have strongly confirmed MRT’s excellent ability in transferring robustness.

5.4 Chapter Summary

This chapter has employed knowledge transfer to enhance IC system robustness in defending against adversarial examples while fighting data scarcity. We have proposed a meta-learning based adversarial defense framework named Meta Robustness Transfer (MRT) empowered by a non-parametric Robust Feature Selection Layer (RFSL) to perform a few-shot robustness transfer. Guided by a robust adversarially-trained (source) IC system, MRT empowers a non-robust (target) IC system to embed samples into robust features without conducting adversarial training; it enables a robust sample classification via a simple similarity comparison, avoiding the need for extensive labeled data. Consequently, MRT can enhance the robustness of the target IC system even in a deficient data regime. Through comparison experiments in Mini-ImageNet, Caltech-UCSD Birds, and CIFAR-FS, we have demonstrated that MRT outperforms its closely competent in transferring robustness across tasks with different system architectures, different labels spaces, and different datasets in extremely low data regimes.

Chapter 6

Conclusion

In this thesis, we have presented deep-driven computer vision techniques as a viable step towards enhancing the reliability of the Image Classification (IC) system and Radio-Frequency Based Human Activity Recognition (RF-HAR) system that built on neural network in the face of two risks - adversarial attacks and data scarcity. This thesis has been written from both a theoretical and an implementation perspective. First, we have proposed a novel adversarial defense framework to boost the classification system's robustness in the face of adversarial attacks. The objective of the research work is to delve further into adversarial attacks and their countermeasures. Following that, we have studied the threat of data scarcity and utilized knowledge transfer via meta-learning to address performance degradation caused by data scarcity. Subsequently, we have merged adversarial defense with knowledge transfer via meta-learning to overcome these two types of risks simultaneously after studying the first two approaches and these two types of risk.

Specifically, at the beginning of this thesis, we have provided an overview of both the IC system and RF-HAR system, and subsequently, we have described the threat of these two risks to systems: adversarial attacks can cheat IC system and RF-HAR system cannot achieve promising performance when it performs environment adaptation in data scarcity. Afterward, based on the study of adversarial defense and knowledge transfer, we have presented our methodologies to achieve the following system reliability enhancement objectives. This thesis focuses on presenting research efforts via deep-driven computer vision techniques for enhancing the reliability of neural network-based classification sys-

tems in the face of the two risks - adversarial attacks and data scarcity. Adversarial defense and knowledge transfer are the main research directions to be investigated.

Beginning with the objective of enhancing the robustness of IC system when it is attacked by adversarial examples, we have provided an overview of existing adversarial defense strategies. After analyzing their strengths and weaknesses, we have given attention to the direct design of robust frameworks, which are more poorly researched. Subsequently, we have presented a novel attack-agnostic adversarial defense framework named Feature Pyramid Decoder (FPD) to enhance the block-based IC system's intrinsic robustness directly. FPD framework implants a target IC system with both denoising and image restoration modules to filter an input image at multiple levels. Furthermore, we have analyzed that network with Softmax as a classification layer's activation function can be pretty sensitive to some perturbations. Consequently, we have deployed a Lipschitz Constant Constraint at the classification layer to limit the output variation in the face of attack perturbation. Moreover, to strategically training the enhanced IC system, we have devised a novel two-phase strategy by employing noisy neighborhood images with multi-task and self-supervised learning. We have investigated the best structure of our FPD and confirmed settings for training strategy through the exploration experiments. Moreover, through a series of comparison experiments under various well-known white-box attacks and black-box attacks on MNIST, the Street View House Numbers (SVHN), and CALTECH datasets, results have demonstrated that the FPD framework can enhance the intrinsic robustness of IC system in an attack-agnostic manner. Furthermore, under adversarial training, the FPD-enhanced IC system has shown better performance than the non-enhanced version.

Next, to improve RF-HAR system environment adaptation performance in data scarcity, we have described emerging existing solutions in two main research directions: environment invariant systems and knowledge transfer, and then discussed their merit and demerit. Moreover, we have found that existing RF-HAR systems only utilize partial information provided by RF signals and motivated us to design a unified framework to handle representative RF sensing techniques. Subsequently, we have employed knowledge transfer via meta-learning for relieving the labeling stress for the RF-HAR system being adaptive to new environments. We have designed RF-Net, a unified meta-learning

based knowledge transfer framework for one-shot RF-based HAR; it contributes to the capability of being fast adapted to new environments with very few labeled samples. We have innovated two designs to constitute RF-Net. A dual-path base HAR network has been designed to learn powerful RF features from both time and frequency domains. It applies to representative RF sensing datasets. A metric-based meta-learning framework has been proposed to endow fast adaptation capability to the HAR network via transferring knowledge from seen environments, enabling base HAR network that transforms inputs into a representation suitable for classification via the similarity comparison allows environment adaptation with very few labeled data. We have conducted hyperparameter searching to explore the best framework structures along with training settings. We have performed extensive experiments on three representative RF sensing techniques, Wi-Fi radio, frequency-modulated continuous wave radio, and impulse radio, in multiple real-world. These experiments have demonstrated that our proposed RF-Net has the capability of being adaptive to a new environment with limited labeled data, and it outperforms state-of-the-art baselines.

After researching the first two approaches and these two types of risk, we have been motivated to perform adversarial defenses with knowledge transfer through meta-learning to solve these two types of risks concurrently. Inspired by the property that knowledge transfer based learning systems can transfer knowledge from known tasks to new tasks and the potential to fight data scarcity, we have deemed “system robustness” as a type of knowledge. This knowledge can be transferred from a robust adversarially-trained (source) IC system to a non-robust (target) IC system, and consequently enhanced robustness of the target IC system. We have described related studies that utilize this idea and discussed their pros and cons. We have found that a gap exists to utilize knowledge transfer and adversarial defense to defend against both adversarial attacks in the data scarcity scenario. Subsequently, we have proposed a meta-learning based adversarial defense framework Meta Robustness Transfer (MRT), to perform a few-shot knowledge transfer in the form of robust features. We have designed a non-parametric Robust Feature Selection Layer (RFSL) to characterize the distribution of robust features. Empowered by RFSL, MRT essentially enables a non-robust system to acquire robust feature embedding from a robust system, avoiding the need for extensive labeled data and re-conducting adversarial training. Consequently, the robustness of the target IC system

can be improved. Through comparison experiments in Mini-ImageNet, Caltech-UCSD Birds, and CIFAR-FS, we have demonstrated that MRT outperforms its closely competent in transferring robustness across tasks with different system architectures, different labels spaces, and different datasets in extremely low data regimes.

Appendix A

Mathematical Proofs in Chapter 3

A.1 Proof: Lipschitz Constant Constraint for Fully-Connected Network

Theorem 2 (the constraint on Lipschitz constant for fully-connected network)

Let NN_{FC} be a N^{class} -way- N^{L} -layer-fully-connected network, $\text{NN}_{\text{FC}}(\mathbf{x})_k$ be the k -th component of the network output given input $\mathbf{x}$, w_i be the weight matrix of the i -th layer of the network, and b_i be a bias matrix of the same layer. Given a noise vector δ , we can bound the variation $\mathcal{V}$ component-wisely from above by:

$$\mathcal{V}_k = |\text{NN}_{\text{FC}}(\mathbf{x})_k - \text{NN}_{\text{FC}}(x + \delta)_k| \leq \frac{e^{\theta_k|_x}(e^\eta - e^{-\eta})}{\sum_p e^{\theta_p|_{x+\delta}}}, \quad (\text{A.1})$$

$$\eta = \max_{k=1, \dots, N^{\text{class}}} \{[w_{N^{\text{L}}} C^{N^{\text{L}}-1} |w_{N^{\text{L}}-1} w_{N^{\text{L}}-2} \dots w_1 \delta| + b_{N^{\text{L}}}]_k\}, \quad (\text{A.2})$$

where $\theta_k|_x$ is the k -th component of the input to Softmax σ_x given input $\mathbf{x}$, a stands for the activation function of earlier layers, C means a 's Lipschitz constant.

Proof:

For N^{class} classify mission, input $\mathbf{x}$ has n dim, fixed activation function is a for each layer, and the network has $N^{\text{L}} + 1$ layers. For simplicity, the assumption is that target network f is a fully connected network and $\mathbf{x}$ is normalized into $\mathbf{x} \in [0, 1]^{n \times 1}$. Consequently, its structure is summarized in Table A.1. Afterward, a noise vector $\delta^{n \times 1}$ is added to $\mathbf{x}$

Table A.1: The structure of the network.

layer	input	Output
0	$\mathbf{x}^{n \times 1}$	/
1	$\mathbf{x}^{n \times 1}$	$a(w_1^{n_1 \times n} \mathbf{x} + b_1^{n_1 \times 1}) = \text{Layer}_1^{n_1 \times 1}$
2	$a(w_1^{n_1 \times n} \mathbf{x} + b_1^{n_1 \times 1}) = \text{Layer}_1^{n_1 \times 1}$	$a(w_2^{n_2 \times n_1} \text{Layer}_1^{n_1 \times 1} + b_2^{n_2 \times 1}) = \text{Layer}_2^{n_2 \times 1}$
...	...	...
i	$a(w_{i-1}^{n_{i-1} \times n_{i-2}} \text{Layer}_{i-2}^{n_{i-2} \times 1} + b_{i-1}^{n_{i-1} \times 1}) = \text{Layer}_{i-1}^{n_{i-1} \times 1}$	$a(w_i^{n_i \times n_{i-1}} \text{Layer}_{i-1}^{n_{i-1} \times 1} + b_i^{n_i \times 1}) = \text{Layer}_i^{n_i \times 1}$
...	...	...
N^L	$a(w_{N^L-1}^{n_{N^L-1} \times n_{N^L-2}} \text{Layer}_{N^L-2}^{n_{N^L-2} \times 1} + b_{N^L-1}^{n_{N^L-1} \times 1}) = \text{Layer}_{N^L-1}^{n_{N^L-1} \times 1}$	$a(w_{N^L}^{k \times n_{N^L-1}} \text{Layer}_{N^L-1}^{n_{N^L-1} \times 1} + b_{N^L}^{k \times 1}) = \text{Layer}_{N^L}^{k \times 1}$
$N^L + 1$	$a(w_{N^L}^{N^{\text{class}} \times n_{N^L-1}} \text{Layer}_{N^L-1}^{n_{N^L-1} \times 1} + b_{N^L}^{N^{\text{class}} \times 1}) = \text{Layer}_{N^L}^{N^{\text{class}} \times 1}$	$\sigma_x(\text{Layer}_{N^L}^{N^{\text{class}} \times 1}) = \text{Layer}_{N^L+1}^{N^{\text{class}} \times 1}$

For the convenience of proof, let $N^L = 2$, and we have:

$$\text{Layer}_2(\mathbf{x}) = a(w_2 a(w_1 \mathbf{x} + b_1) + b_2), \quad (\text{A.3})$$

$$\text{Layer}_2(\mathbf{x} + \delta) = a(w_2 a(w_1(\mathbf{x} + \delta) + b_1) + b_2), \quad (\text{A.4})$$

$$\text{Layer}_3(\mathbf{x})_k = \frac{\exp(a(w_2 a(w_1 \mathbf{x} + b_1) + b_2)_k)}{\sum_i \exp(a(w_2 a(w_1 \mathbf{x} + b_1) + b_2)_i)}, \quad (\text{A.5})$$

$$\text{Layer}_3(\mathbf{x} + \delta)_k = \frac{\exp(a(w_2 a(w_1(\mathbf{x} + \delta) + b_1) + b_2)_k)}{\sum_i \exp(a(w_2 a(w_1(\mathbf{x} + \delta) + b_1) + b_2)_i)}. \quad (\text{A.6})$$

Define D_k as the k -th component of the variation:

$$D_k = \text{Layer}_3(\mathbf{x})_k - \text{Layer}_3(\mathbf{x} + \delta)_k, \quad (\text{A.7})$$

$$= \frac{\exp(a(w_2 a(w_1 \mathbf{x} + b_1) + b_2)_k) \times \sum \exp(a(w_2 a(w_1(\mathbf{x} + \delta) + b_1) + b_2)_i)}{\sum \exp(a(w_2 a(w_1 \mathbf{x} + b_1) + b_2)_i) \times \sum \exp(a(w_2 a(w_1(\mathbf{x} + \delta) + b_1) + b_2)_i)} \quad (\text{A.8})$$

$$- \frac{\exp(a(w_2 a(w_1(\mathbf{x} + \delta) + b_1) + b_2)_k) \times \sum \exp(a(w_2 a(w_1 \mathbf{x} + b_1) + b_2)_i)}{\sum \exp(a(w_2 a(w_1 \mathbf{x} + b_1) + b_2)_i) \times \sum \exp(a(w_2 a(w_1(\mathbf{x} + \delta) + b_1) + b_2)_i)}, \quad (\text{A.9})$$

$$\mathcal{V}_k = |D_k|. \quad (\text{A.10})$$

Firstly, the relationship between $\text{Layer}_2(\mathbf{x})$ and $\text{Layer}_2(\mathbf{x} + \delta)$ is analyzed. Supposed that activation function a has property satisfies:

$$|a(\mathbf{x}) - a(\mathbf{y})| \leq C |\mathbf{x} - \mathbf{y}|, \quad (\text{A.11})$$

where C is known as Lipschitz constant.

Therefore, following formulation can be computed:

$$|a(w_1 \mathbf{x} + b_1) - a(w_1(\mathbf{x} + \delta) + b_1)| \leq C |w_1 \delta|. \quad (\text{A.12})$$

The $\leq$ means every corresponding element from left and right satisfies the inequality.

Subsequently, this property is applied to Layer_2

$$|\text{Layer}_2(\mathbf{x} + \delta) - \text{Layer}_2(\mathbf{x})| \quad (\text{A.13})$$

$$= |a(w_2 a(w_1(\mathbf{x} + \delta) + b_1) + b_2) - a(w_2 a(w_1 \mathbf{x} + b_1) + b_2)| \quad (\text{A.14})$$

$$\leq C |w_2 (a(w_1(\mathbf{x} + \delta) + b_1) - a(w_1 \mathbf{x} + b_1))| \leq C^2 |w_2 w_1 \delta|. \quad (\text{A.15})$$

Defining $\zeta = \max\{[C^2 |w_2 w_1 \delta|]_i\}$ means the biggest absolute value of element in vector $C^2 w_2 w_1 \delta$. Now we can estimate the upper and lower bound of D_k :

$$D_k = \frac{\exp(a(w_2 a(w_1 \mathbf{x} + b_1) + b_2)_k) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.16})$$

$$- \frac{\exp(a(w_2 a(w_1(\mathbf{x} + \delta) + b_1) + b_2)_k) \times \sum \exp(\text{Layer}_2(\mathbf{x})_i)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.17})$$

$$\leq \frac{\exp(\text{Layer}_2(\mathbf{x})_k) \times \sum \exp(\text{Layer}_2(\mathbf{x})_i + C^2 |w_2 w_1 \delta|)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.18})$$

$$- \frac{\exp(\text{Layer}_2(\mathbf{x})_k - C^2 |w_2 w_1 \delta|) \times \sum \exp(\text{Layer}_2(\mathbf{x})_i)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.19})$$

$$\leq \frac{\exp(\text{Layer}_2(\mathbf{x})_k) \times \sum \exp(\text{Layer}_2(\mathbf{x})_i + \zeta)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.20})$$

$$- \frac{\exp(\text{Layer}_2(\mathbf{x})_k - \zeta) \times \sum \exp(\text{Layer}_2(\mathbf{x})_i)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.21})$$

$$= \frac{\exp(\text{Layer}_2(\mathbf{x})_k) \times (\exp(\zeta) - \exp(-\zeta))}{\sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)}, \quad (\text{A.22})$$

and

$$D_k = \frac{\exp(f(w_2 f(w_1 \mathbf{x} + b_1) + b_2)_k) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.23})$$

$$- \frac{\exp(a(w_2 f(w_1(\mathbf{x} + \delta) + b_1) + b_2)_k) \times \sum \exp(\text{Layer}_2(\mathbf{x})_i)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.24})$$

$$\geq \frac{\exp(\text{Layer}_2(\mathbf{x})_k) \times \sum \exp(\text{Layer}_2(\mathbf{x})_i - C^2 |w_2 w_1 \delta|)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.25})$$

$$- \frac{\exp(\text{Layer}_2(\mathbf{x})_k + C^2 |w_2 w_1 \delta|) \times \sum \exp(\text{Layer}_2(\mathbf{x})_i)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.26})$$

$$\geq \frac{\exp(\text{Layer}_2(\mathbf{x})_k) \times \sum \exp(\text{Layer}_2(\mathbf{x})_i - \zeta)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.27})$$

$$- \frac{\exp(\text{Layer}_2(\mathbf{x})_k + \zeta) \times \sum \exp(\text{Layer}_2(\mathbf{x})_i)}{\sum \exp(\text{Layer}_2(\mathbf{x})_i) \times \sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)} \quad (\text{A.28})$$

$$= \frac{\exp(\text{Layer}_2(\mathbf{x})_k) \times (\exp(-\zeta) - \exp(\zeta))}{\sum \exp(\text{Layer}_2(\mathbf{x} + \delta)_i)}. \quad (\text{A.29})$$

The limitation of $\mathcal{V}_k$:

$$\lim_{\zeta \rightarrow 0} \mathcal{V}_k = 0.$$

To estimate the bound of $|w_2 w_1 \delta|$, weights matrix (except for the w_1 and w_L) is filled by zero to keep shape consistency $m \times m$:

$$w_1 = \begin{bmatrix} w_{1,1} & \dots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{m,1} & \dots & w_{m,m} \end{bmatrix},$$

$$\delta = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_n \end{bmatrix},$$

$$w_2 = \begin{bmatrix} w'_{1,1} & \dots & w'_{1,m} \\ \vdots & \ddots & \vdots \\ w'_{m,1} & \dots & w'_{m,m} \end{bmatrix},$$

$$w_1 \delta = \begin{bmatrix} \sum_{j=1}^n w_{1,j} \times \delta_j \\ \vdots \\ \sum_{j=1}^n w_{m,j} \times \delta_j \end{bmatrix}$$

and

$$w_2 w_1 \delta = \begin{bmatrix} \sum_{i=1}^m (w'_{1,i} \times \sum_{j=1}^n w_{i,j} \times \delta_j) \\ \vdots \\ \sum_{i=1}^m (w'_{m,i} \times \sum_{j=1}^n w_{i,j} \times \delta_j) \end{bmatrix}.$$

The bound could be roughly estimated by using the biggest element of weight matrix:

$$w_1 \delta \leq \begin{bmatrix} \max_{i=1,2,\dots,n} \{[\delta]_i\} \times \sum_{j=1}^n w_{1,j} \\ \vdots \\ \max_{i=1,2,\dots,n} \{[\delta]_i\} \times \sum_{j=1}^n w_{m,j} \end{bmatrix} \quad (\text{A.30})$$

$$\leq \begin{bmatrix} \max_{i=1,2,\dots,n} \{[\delta]_i\} \times \max_{i=1,2,\dots,m} \{[\sum_{j=1}^n w_{i,j}]_i\} \\ \vdots \\ \max_{i=1,2,\dots,n} \{[\delta]_i\} \times \max_{i=1,2,\dots,m} \{[\sum_{j=1}^n w_{i,j}]_i\} \end{bmatrix} \quad (\text{A.31})$$

and

$$w_2 w_1 \delta \leq \quad (\text{A.32})$$

$$\begin{bmatrix} \max_{i=1,2,\dots,n} \{[\delta]_i\} \\ \times \max_{i=1,2,\dots,m} \{[\sum_{j=1}^n w_{i,j}]_i\} \times \max_{i=1,2,\dots,m} \{[\sum_{j=1}^m w'_{i,j}]_i\} \\ \vdots \\ \max_{i=1,2,\dots,n} \{[\delta]_i\} \\ \times \max_{i=1,2,\dots,m} \{[\sum_{j=1}^n w_{i,j}]_i\} \times \max_{i=1,2,\dots,m} \{[\sum_{j=1}^m w'_{i,j}]_i\} \end{bmatrix}. \quad (\text{A.33})$$

$\rho_i = \max_{q=1,2,\dots,m} \{[\sum_j w_{i,k,j}]_q\}$ is defined to represent the biggest one in layer i weight matrix summing up by row, similar as the 1-matrix norm but without using element absolute value.

Therefore, Eqn. (A.32) could be further extended as follows:

$$w_{N^L} w_{N^L-1} \dots w_1 \delta \leq \begin{bmatrix} \sum_{i=1}^{N^L} \rho_i \times \max_{j=1,2,\dots,n} \{[\delta]_j\} \\ \vdots \\ \sum_{i=1}^{N^L} \rho_i \times \max_{j=1,2,\dots,n} \{[\delta]_j\} \end{bmatrix}.$$

So

$$|\text{Layer}_{N^L}(\mathbf{x} + \delta) - \text{Layer}_{N^L}(\mathbf{x})|, \quad (\text{A.34})$$

$$\leq \left(C \times \max_{j=1,2,\dots,N^L} \{[\rho_i]_j\} \right)^{N^L} \times \max_{j=1,2,\dots,n} \{[\delta]_j\}. \quad (\text{A.35})$$

If $C < \frac{1}{\max_{j=1,2,\dots,N^L}\{[\rho_i]_j\}}$, then $\lim_{C \rightarrow 0} \mathcal{V}_k = 0$ or equally $\lim_{N^L \rightarrow \infty} \mathcal{V}_k = 0$.

Assumed that every matrix element satisfies $w_{i,j,k} \sim N(\mu_i, \sigma_i)$, the expectation and variance of $\sum_j w_{i,k,j}$ could be computed:

$$E \left[\sum_j w_{i,k,j} \right]_{w_{i,k,j} \sim P(\mu_i, \sigma_i)} = m\mu_i, \quad (\text{A.36})$$

$$\text{Var} \left[\sum_j w_{i,k,j} \right]_{w_{i,k,j} \sim P(\mu_i, \sigma_i)} = m\sigma_i^2. \quad (\text{A.37})$$

Consequently,

$$\frac{\sum_j w_{i,k,j} - m\mu_i}{m\sigma_i^2} \sim N(0, 1).$$

means that ρ_i will not be bigger than $m\mu_i$ if σ_i^2 is small.

Comparing with a network without involving an activation function before Softmax σ_x at layer N^L , such as

$$\sigma_x(w_{N^L} \text{Layer}_{N^L-1}(\mathbf{x}) + b_{N^L})$$

and

$$\theta_k|_x = [w_{N^L} a(w_{N^L-1}(\dots(a(w_1 x + b_1))) + b_{N^L-1}) + b_{N^L}]_k,$$

$$\theta_k|_{x+\delta} = [w_{N^L} a(w_{N^L-1}(\dots(a(w_1(x + \delta) + b_1))) + b_{N^L-1}) + b_{N^L}]_k,$$

The bound could be computed:

$$\mathcal{V}_k \leq \frac{e^{\theta_k|_x} (e^\eta - e^{-\eta})}{\sum_p e^{\theta_p|_{x+\delta}}},$$

and

$$\eta = \max_{k=1,2,\dots,N^{\text{class}}} \{[w_{N^L} C^{N^L-1} |w_{N^L-1} w_{N^L-2} \dots w_1 \delta| + b_{N^L}]_k\}.$$

If let $\tau = \sqrt{(N^L-1) \max_{k=1,2,\dots,N^{\text{class}}} \{[|w_{N^L-1} w_{N^L-2} \dots w_1 \delta|]_k\}}$, then three situations could be considered:

1.

$$\lim_{0 \leq C\tau < 1, 0 < N^{\text{class}} < M, N^L \rightarrow \infty} \eta = \max_{k=1,2,\dots,N^{\text{class}}} \{[b_{N^L}]_k\}$$

2.

$$\lim_{C\tau=1, 0 < N^{\text{class}} < M, N^L \rightarrow \infty} \eta = \max_{k=1,2,\dots,N^{\text{class}}} \{[\rho_{N^L} + b_{N^L}]_k\}$$

3.

$$\lim_{C\tau > 1, 0 < N^{\text{class}} < M, N^L \rightarrow \infty} \eta \rightarrow \infty$$

Therefore, if $C\tau \leq 1$, w_{N^L} and b_{N^L} will have a bigger influence than $C^{N^L-1} |w_{N^L-1} w_{N^L-2} \dots w_1 \delta|$.

Appendix B

Experimental Results in Chapter 5

B.1 Input Gradient Adversarial Matching Performance

We mainly evaluate Input Gradient Adversarial Matching Performance (IGAM) under two experiment settings: transfer robustness from robust Resnet18 trained on CIFAR-FS (Mini-ImageNet) to Resnet18 on Mini-ImageNet (CIFAR-FS). As displayed in Table B.1, we observe that IGAM cannot achieve comparable performance on clean samples and its performance is almost random guess level. In other words, given limited samples, IGAM fails to train a target system which has an acceptable generalization capability. Classifier with good generalization performance is an essential precondition for being attacked.

Table B.1: IGAM performance on clean samples. MI and FS are Mini-ImageNet and CIFAR-FS.

Task	Res18(MI)	Res18(FS)
3-way-1-shot	33.960	33.300
3-way-5-shot	33.610	34.320
5-way-1-shot	20.140	19.630
5-way-5-shot	19.990	19.330

B.2 Transfer Robustness across Systems and Datasets

Table B.2: Accuracy (%) of Mobilenet as f_θ under 5-way 1-shot tasks on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	26.460	0.090	0.000	12.530	0.000
AQ	22.600	3.989	0.060	16.540	0.000
MRT	23.670	8.728	4.629	14.550	21.700

Table B.3: Accuracy (%) of Mobilenet as f_θ under 5-way 5-shot tasks on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	43.020	0.000	0.000	8.806	0.000
AQ	28.130	2.900	1.078	29.030	0.100
MRT	30.660	4.300	1.101	10.280	11.640

Table B.4: Accuracy (%) of Resnet34 as f_θ under 5-way 1-shot tasks on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	35.500	0.010	0.000	0.100	0.011
AQ	23.160	0.030	0.283	1.806	22.940
MRT	23.580	5.070	3.856	9.011	20.200

Table B.5: Accuracy (%) of Resnet34 as f_θ under 5-way 5-shot tasks on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	56.570	0.000	0.000	0.000	0.000
AQ	26.730	7.844	6.683	12.140	22.040
MRT	31.030	6.256	4.980	9.408	26.430

Table B.6: Accuracy (%) of Resnet18 as f_θ under 5-way 1-shot tasks on CIFAR-FS.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	29.010	3.939	3.178	7.844	3.710
AQ	22.890	11.820	11.540	13.920	12.290
MRT	23.930	8.895	8.050	12.540	22.040

Table B.7: Accuracy (%) of Resnet18 as f_θ under 5-way 5-shot tasks on CIFAR-FS.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	55.960	0.833	0.673	4.067	0.905
AQ	25.160	12.620	12.280	17.910	12.470
MRT	42.790	7.589	6.078	14.040	27.890

Table B.8: Accuracy (%) of Resnet18 as f_θ under 5-way 1-shot tasks on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	38.310	0.000	0.000	0.000	0.000
AQ	22.180	1.606	1.161	3.589	14.980
MRT	23.240	6.872	5.922	11.560	22.400

Table B.9: Accuracy (%) of Resnet18 as f_θ under 5-way 5-shot tasks on Mini-ImageNet.

Name	Clean	PGD-7	PGD-40	FGSM	C&W
NT	47.890	0.000	0.000	0.000	0.000
AQ	27.710	6.661	6.072	9.256	19.140
MRT	35.530	2.161	1.245	7.828	24.730

B.3 Time Complexity

(a) Training time.

(b) Total attacking time of performing all four attacks.

Figure B.1: Time complexity comparison, given Resnet18 as f_θ on Mini-ImageNet.

References

- [1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *Proc. of the ICLR*, 2015.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proc. of the CVPR*, 2016.
- [4] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, “Deep Convolutional Neural Networks on Multi-channel Time Series for Human Activity Recognition,” in *Proc. of the IJCAI*, vol. 15. Buenos Aires, Argentina, 2015, pp. 3995–4001.
- [5] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in *Proc. of the NIPS*, 2015, pp. 91–99.
- [6] H. Noh, S. Hong, and B. Han, “Learning Deconvolution Network for Semantic Segmentation,” in *Proc. of the ICCV*, 2015, pp. 1520–1528.
- [7] J. Zhang, W. Wang, C. Lu, J. Wang, and A. K. Sangaiah, “Lightweight Deep Network for Traffic Sign Classification,” *Annals of Telecommunications*, vol. 75, no. 7, pp. 369–379, 2020.
- [8] P. Rasti, T. Uiboupin, S. Escalera, and G. Anbarjafari, “Convolutional Neural Network Super Resolution for Face Recognition in Surveillance Monitoring,” in *Proc. of the AMDO*. Springer, 2016, pp. 175–184.
- [9] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” *Proc. of the ICLR*, 2015.
- [10] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, “HotFlip: White-Box Adversarial Examples for Text Classification,” in *Proc. of the ACL*, 2018, pp. 31–36.
- [11] D. Zügner, A. Akbarnejad, and S. Günnemann, “Adversarial Attacks on Neural Networks for Graph Data,” in *Proc. of the KDD*, 2018, pp. 2847–2856.
- [12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing Properties of Neural Networks,” *Proc. of the ICLR*, 2014.
- [13] A. Nguyen, J. Yosinski, and J. Clune, “Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images,” in *Proc. of the CVPR*, 2015, pp. 427–436.
- [14] Z. Che, A. Borji, G. Zhai, S. Ling, G. Guo, and P. Le Callet, “Adversarial Attacks Against Deep Saliency Models,” *CoRR*, vol. abs/1904.01231, 2019.
- [15] K. Xu, S. Liu, G. Zhang, M. Sun, P. Zhao, Q. Fan, C. Gan, and X. Lin, “Interpreting Adversarial Examples by Activation Promotion and Suppression,” *CoRR*, vol. abs/1904.02057, 2019.
- [16] T. W. Hnat, V. Srinivasan, J. Lu, T. I. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse, “The Hitchhiker’s Guide to Successful Residential Sensing Deployments,” in *Proc. of the 9th ACM SenSys*, 2011, pp. 232–245.
- [17] A. Jalal, M. Z. Uddin, and T.-S. Kim, “Depth Video-Based Human Activity Recognition System Using Translation and Scaling Invariant Features for Life Logging at Smart Home,” *IEEE Transaction on Consumer Electronics*, vol. 58, no. 3, pp. 863–871, 2012.

REFERENCES

- [18] Z. Chi, Y. Yao, T. Xie, X. Liu, Z. Huang, W. Wang, and T. Zhu, “EAR: Exploiting Uncontrollable Ambient RF Signals in Heterogeneous Networks for Gesture Recognition,” in *Proc. of the 16th ACM SenSys*, 2018, pp. 237–249.
- [19] T. Hao, C. Bi, G. Xing, R. Chan, and L. Tu, “MindfulWatch: A Smartwatch-Based System For Real-Time Respiration Monitoring During Meditation,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, Sep. 2017. [Online]. Available: <https://doi.org/10.1145/3130922>
- [20] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles, “PBN: Towards Practical Activity Recognition Using Smartphone-Based Body Sensor Networks,” in *Proc. of the 9th ACM SenSys*, 2011, pp. 246–259.
- [21] A. Ghose, K. Chakravarty, A. K. Agrawal, and N. Ahmed, “Unobtrusive Indoor Surveillance of Patients at Home Using Multiple Kinect Sensors,” in *Proc. of the 11st ACM SenSys*, 2013, pp. 1–2.
- [22] M. Humenberger, S. Schraml, C. Sulzbachner, A. N. Belbachir, A. Srp, and F. Vajda, “Embedded Fall Detection with A Neural Network and Bio-Inspired Stereo Vision,” in *Proc. of the 25th IEEE CVPR Workshops*, 2012, pp. 60–67.
- [23] Y. Wang, K. Wu, and L. M. Ni, “WiFall: Device-Free Fall Detection by Wireless Networks,” in *Proc. of the 33rd IEEE INFOCOM*, 2014, pp. 271–279.
- [24] Y. Tian, G.-H. Lee, H. He, C.-Y. Hsu, and D. Katabi, “RF-Based Fall Monitoring Using Convolutional Neural Networks,” in *Proc. of the 18th ACM UbiComp*, 2018, pp. 137:1–24.
- [25] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, and et al., “Towards Environment Independent Device Free Human Activity Recognition,” in *Proc. of the MobiCom*, 2018, pp. 289–304.
- [26] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, “Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi,” in *Proc. of the MobiSys*, 2019, pp. 313–325.
- [27] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, “Thermometer Encoding: One Hot Way to Resist Adversarial Examples,” in *Proc. of the ICLR*, 2018.
- [28] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, “Countering Adversarial Images Using Input Transformations,” in *Proc. of the ICLR*, 2018.
- [29] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, “Mitigating Adversarial Effects Through Randomization,” in *Proc. of the ICLR*, 2018.
- [30] T. Luo, T. Cai, M. Zhang, S. Chen, and L. Wang, “Random Mask: Towards Robust Convolutional Neural Networks,” *Proc. of the ICLR*, 2019.
- [31] P. Samangouei, M. Kabkab, and R. Chellappa, “Defense-GAN: Protecting Classifiers against Adversarial Attacks Using Generative Models,” in *Proc. of the ICLR*, 2018.
- [32] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, “Feature Denoising for Improving Adversarial Robustness,” in *Proc. of the CVPR*, 2019, pp. 501–509.
- [33] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, “Evaluating the Robustness of Neural Networks: an Extreme Value Theory Approach,” in *Proc. of the ICLR*, 2018.
- [34] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards Deep Learning Models Resistant to Adversarial Attacks,” in *Proc. of the ICLR*, 2018.
- [35] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang, “CrossSense: Towards Cross-Site and Large-Scale WiFi Sensing,” in *Proc. of the MobiCom*, 2018, pp. 305–320.
- [36] T. Gong, Y. Kim, J. Shin, and S.-J. Lee, “MetaSense: Few-Shot Adaptation to Untrained Conditions in Deep Mobile Sensing,” in *Proc. of the 17th ACM Sensys*, 2019, pp. 110–123.
- [37] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, “Meta-Learning with Latent Embedding Optimization,” *Proc. of the 7th ICLR*, 2019.

REFERENCES

- [38] A. Shafahi, P. Saadatpanah, C. Zhu, A. Ghiasi, C. Studer, D. Jacobs, and T. Goldstein, “Adversarially Robust Transfer Learning,” in *Proc. of the ICLR*, 2019.
- [39] A. Chan, Y. Tay, and Y.-S. Ong, “What it Thinks is Important is Important: Robustness Transfers through Input Gradients,” in *Proc. of the CVPR*, 2020, pp. 332–341.
- [40] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, “Robustness of classifiers: from adversarial to random noise,” in *NIPS*, 2016, pp. 1632–1640.
- [41] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, and S. Soatto, “Analysis of Universal Adversarial Perturbations,” *arXiv preprint arXiv:1705.09554*, 2017.
- [42] A. Fawzi, O. Fawzi, and P. Frossard, “Analysis of Classifiers’ Robustness to Adversarial Perturbations,” *Machine Learning*, vol. 107, no. 3, pp. 481–508, 2018.
- [43] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein, “Are Adversarial Examples Inevitable?” *arXiv preprint arXiv:1809.02104*, 2018.
- [44] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, “Adversarial Examples are Not Bugs, They are Features,” in *Proc. of the NIPS*, 2019, pp. 125–136.
- [45] Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into Transferable Adversarial Examples and Black-Box Attacks,” in *Proc. of the ICLR*, 2017.
- [46] N. Papernot, P. McDaniel, and I. Goodfellow, “Transferability in Machine Learning: from Phenomena to Black-Box Attacks Using Adversarial Samples,” *arXiv preprint arXiv:1605.07277*, 2016.
- [47] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “The Space of Transferable Adversarial Examples,” *arXiv preprint arXiv:1704.03453*, 2017.
- [48] B. Biggio, B. Nelson, and P. Laskov, “Poisoning Attacks Against Support Vector Machines,” in *Proc. of the ICML*, 2012, pp. 1467–1474.
- [49] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, “Evasion Attacks Against Machine Learning at Test Time,” in *Proc. of the ECML PKDD*. Springer, 2013, pp. 387–402.
- [50] D. C. Liu and J. Nocedal, “On the Limited Memory BFGS Method for Large Scale Optimization,” *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.
- [51] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial Machine Learning at Scale,” in *Proc. of the ICLR*, 2017.
- [52] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: a Simple and Accurate Method to Fool Deep Neural Networks,” in *Proc. of the CVPR*, 2016, pp. 2574–2582.
- [53] N. Carlini and D. Wagner, “Towards Evaluating the Robustness of Neural Networks,” in *Proc. of the S&P*, 2017, pp. 39–57.
- [54] N. Carlini, G. Katz, C. Barrett, and D. L. Dill, “Provably Minimally-Distorted Adversarial Examples,” *arXiv preprint arXiv:1709.10207*, 2017.
- [55] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical Black-Box Attacks Against Machine Learning,” in *Proc. of the AsiaCCS*, 2017, pp. 506–519.
- [56] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, “Boosting Adversarial Attacks with Momentum,” in *Proc. of the CVPR*, 2018, pp. 9185–9193.
- [57] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, “ZOO: Zeroth Order Optimization Based Black-Box Attacks to Deep Neural Networks Without Training Substitute Models,” in *Proc. of the AISec*, 2017, pp. 15–26.
- [58] H. Truong, S. Zhang, U. Muncuk, P. Nguyen, N. Bui, A. Nguyen, Q. Lv, K. Chowdhury, T. Dinh, and T. Vu, “Capband: Battery-Free Successive Capacitance Sensing Wristband for Hand Gesture Recognition,” in *Proc. of the 16th ACM SenSys*, 2018, pp. 54–67.

REFERENCES

- [59] C. Bi, G. Xing, T. Hao, J. Huh-Yoo, W. Peng, M. Ma, and X. Chang, “FamilyLog: Monitoring Family Mealtime Activities by Mobile Devices,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 8, pp. 1818–1830, 2020.
- [60] Y. He, J. Liang, and Y. Liu, “Pervasive Floorplan Generation Based on Only Inertial Sensing: Feasibility, Design, and Implementation,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1132–1140, 2017.
- [61] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, “Jointly Learning Heterogeneous Features for RGB-D Activity Recognition,” in *Proc. of the 28th IEEE CVPR*, 2015, pp. 5344–5352.
- [62] M. Ma, H. Fan, and K. M. Kitani, “Going Deeper into First-Person Activity Recognition,” in *Proc. of the 29th IEEE CVPR*, 2016, pp. 1894–1903.
- [63] X. Liu, P. Ghosh, O. Ulutan, B. Manjunath, K. Chan, and R. Govindan, “Caesar: Cross-Camera Complex Activity Recognition,” in *Proc. of the 17th ACM SenSys*, 2019, pp. 232–244.
- [64] T. Hao, G. Xing, and G. Zhou, “iSleep: Unobtrusive Sleep Quality Monitoring Using Smartphones,” in *Proc. of the 11st ACM SenSys*, 2013, pp. 1–14.
- [65] X. Xu, J. Yu, Y. Chen, Y. Zhu, L. Kong, and M. Li, “BreathListener: Fine-Grained Breathing Monitoring in Driving Environments Utilizing Acoustic Signals,” in *Proc. of the MobiSys*, 2019, pp. 54–66.
- [66] A. Wang and S. Gollakota, “Millisonic: Pushing the Limits of Acoustic Motion Tracking,” in *Proc. of the CHI*, 2019, pp. 1–11.
- [67] J. Lin, C. Gan, and S. Han, “TSM: Temporal Shift Module for Efficient Video Understanding,” in *Proc. of the 33rd IEEE ICCV*, 2019, pp. 7083–7093.
- [68] X. Zhang, L. Yao, C. Huang, S. Wang, M. Tan, G. Long, and C. Wang, “Multi-Modality Sensor Data Classification with Selective Attention,” in *Proc. of the 27th IJCAI*, 2018, pp. 3111–3117.
- [69] S. Sigg, M. Scholz, S. Shi, Y. Ji, and M. Beigl, “RF-Sensing of Activities from Non-Cooperative Subjects in Device-Free Recognition Systems Using Ambient and Local Signals,” *IEEE Transaction on Mobile Computing*, vol. 13, no. 4, pp. 907–920, 2013.
- [70] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Tool Release: Gathering 802.11n Traces with Channel State Information,” *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 53–53, 2011.
- [71] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, “Understanding and Modeling of WiFi Signal Based Human Activity Recognition,” in *Proc. of the 21st ACM MobiCom*, 2015, pp. 65–76.
- [72] Z. Chen, L. Zhang, C. Jiang, Z. Cao, and W. Cui, “WiFi CSI Based Passive Human Activity Recognition Using Attention Based BLSTM,” *IEEE Transaction on Mobile Computing*, vol. 18, no. 11, pp. 2714–2724, 2018.
- [73] D. Wu, D. Zhang, C. Xu, H. Wang, and X. Li, “Device-Free WiFi Human Sensing: From Pattern-Based to Model-Based Approaches,” *IEEE Communications Magazine*, vol. 55, no. 10, pp. 91–97, 2017.
- [74] M. Zhao, Y. Tian, H. Zhao, M. A. Alsheikh, T. Li, R. Hristov, Z. Kabelac, D. Katabi, and A. Torralba, “RF-Based 3D Skeletons,” in *Proc. of ACM SIGCOMM*, 2018, p. 267–281.
- [75] F. Adib, Z. Kabelac, and D. Katabi, “Multi-Person Localization via RF Body Reflections,” in *Proc. of the 11st USENIX NSDI*, 2015, p. 279–292.
- [76] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, and T. W. Kim, “A Hand Gesture Recognition Sensor Using Reflected Impulses,” *IEEE Sensors Journal*, vol. 17, no. 10, pp. 2975–2976, 2017.
- [77] J. Park and S. H. Cho, “IR-UWB Radar Sensor for Human Gesture Recognition by Using Machine Learning,” in *Proc. of HPCC-SmartCity-DSS*, 2016, pp. 1246–1249.
- [78] T. Zheng, Z. Chen, C. Cai, J. Luo, and X. Zhang, “V²iFi: in-Vehicle Vital Sign Monitoring via Compact RF Sensing,” in *Proc. of the 20th ACM UbiComp*, 2020, pp. 70:1–27.
- [79] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. USA: Cambridge University Press, 2005.
- [80] B. R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*. CRC press, 2002.

REFERENCES

- [81] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-Based Learning Applied to Document Recognition,” *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [82] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet Classification with Deep Convolutional Neural Networks,” in *Proc. of the NIPS*, 2012, pp. 1097–1105.
- [83] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, “SignFi: Sign Language Recognition Using WiFi,” in *Proc. of the UbiComp*, 2018, pp. 23:1–21.
- [84] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, “Keystroke Recognition Using WiFi Signals,” in *Proc. of the MobiCom*, 2015, pp. 90–102.
- [85] R. H. Venkatnarayan, G. Page, and M. Shahzad, “Multi-User Gesture Recognition Using WiFi,” in *Proc. of the MobiSys*, 2018, pp. 401–413.
- [86] A. Virmani and M. Shahzad, “Position and Orientation Agnostic Gesture Recognition Using WiFi,” in *Proc. of the MobiSys*, 2017, pp. 252–264.
- [87] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, “Through-Wall Human Pose Estimation Using Radio Signals,” in *Proc. of the 31st IEEE CVPR*, 2018, pp. 7356–7365.
- [88] H. Abdelnasser, K. Harras, and M. Youssef, “A Ubiquitous WiFi-Based Fine-Grained Gesture Recognition System,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2474–2487, 2019.
- [89] J. McIntosh, A. Marzo, M. Fraser, and C. Phillips, “EchoFlex: Hand Gesture Recognition Using Ultrasound Imaging,” in *Proc. of the CHI*, 2017, pp. 1923–1934.
- [90] G. Li, S. Ding, J. Luo, and C. Liu, “Enhancing Intrinsic Adversarial Robustness via Feature Pyramid Decoder,” in *Proc. of the CVPR*, June 2020.
- [91] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting Adversarial Examples in Deep Neural Networks,” in *Proc. of the NDSS*, 2018.
- [92] A. A. Efros and W. T. Freeman, “Image Quilting for Texture Synthesis and Transfer,” in *Proc. of the SIGGRAPH*, 2001, pp. 341–346.
- [93] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, “A Study of the Effect of JPG Compression on Adversarial Images,” *arXiv preprint arXiv:1608.00853*, 2016.
- [94] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. D. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar, “Stochastic Activation Pruning for Robust Adversarial Defense,” in *Proc. of the ICLR*, 2018.
- [95] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples,” in *Proc. of the ICML*, 2018, pp. 274–283.
- [96] C. Guo, M. Rana, M. Cisse, and L. v. d. Maaten, “Countering Adversarial Images Using Input Transformations,” in *Proc. of the ICLR*, 2018.
- [97] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh, “Towards Robust Neural Networks via Random Self-Ensemble,” in *Proc. of the ECCV*, 2018, pp. 369–385.
- [98] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, “Certified Robustness to Adversarial Examples with Differential Privacy,” in *Proc. of the SP. IEEE*, 2019, pp. 656–672.
- [99] C. Dwork and J. Lei, “Differential Privacy and Robust Statistics,” in *Proc. of the STOC*, 2009, pp. 371–380.
- [100] B. Li, C. Chen, W. Wang, and L. Carin, “Certified Adversarial Robustness with Additive Noise,” in *Proc. of the NIPS*, 2019, pp. 9464–9474.
- [101] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, “PixelDefend: Leveraging Generative Models to Understand and Defend Against Adversarial Examples,” in *Proc. of the ICLR*, 2018.
- [102] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, “Conditional Image Generation with PixelCNN Decoders,” *Advances in neural information processing systems*, vol. 29, pp. 4790–4798, 2016.

REFERENCES

- [103] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Nets,” in *Proc. of the NIPS*, 2014, pp. 2672–2680.
- [104] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, “Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser,” in *Proc. of the CVPR*, 2018, pp. 1778–1787.
- [105] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and Composing Robust Features with Denoising Autoencoders,” in *Proc. of the ICML*, 2008, pp. 1096–1103.
- [106] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Proc. of the MICCAI*. Springer, 2015, pp. 234–241.
- [107] N. Carlini and D. A. Wagner, “Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods,” *CoRR*, vol. abs/1705.07263, 2017. [Online]. Available: <http://arxiv.org/abs/1705.07263>
- [108] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 2012.
- [109] P. L. Bartlett, “For Valid Generalization the Size of the Weights Is More Important Than the Size of the Network,” in *Proc. of the NIPS*, 1997, pp. 134–140.
- [110] M. Hein and M. Andriushchenko, “Formal Guarantees on the Robustness of a Classifier Against Adversarial Manipulation,” in *Proc. of the NIPS*, 2017, pp. 2266–2276.
- [111] A. M. Oberman and J. Calder, “Lipschitz Regularized Deep Neural Networks Converge and Generalize,” *arXiv preprint arXiv:1808.09540*, 2018.
- [112] K. Y. Xiao, V. Tjeng, N. M. M. Shafiullah, and A. Madry, “Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability,” in *Proc. of the ICLR*, 2018.
- [113] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, “Parseval Networks: Improving Robustness to Adversarial Examples,” in *Proc. of the ICML*, 2017, pp. 854–863.
- [114] Y. Tsuzuku, I. Sato, and M. Sugiyama, “Lipschitz-Margin Training: Scalable Certification of Perturbation Invariance for Deep Neural Networks,” *Proc. of the NIPS*, vol. 31, pp. 6541–6550, 2018.
- [115] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial Examples in the Physical World,” in *Proc. of the ICLR Workshop*, 2017.
- [116] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proc. of the ICML Workshop*, 2015, pp. 448–456.
- [117] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “Ensemble Adversarial Training: Attacks and Defenses,” in *Proc. of the ICLR*, 2018.
- [118] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, “Adversarial training for free!” in *Proc. of the NIPS*, 2019, pp. 3358–3369.
- [119] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, “You Only Propagate Once: Painless Adversarial Training Using Maximal Principle,” in *Proc. of the NIPS*, vol. 2, no. 3, 2019.
- [120] Q. Li, S. Haque, C. Anil, J. Lucas, R. B. Grosse, and J.-H. Jacobsen, “Preventing Gradient Attenuation in Lipschitz Constrained Convolutional Networks,” in *Proc. of the NIPS*, 2019, pp. 15 390–15 402.
- [121] C. Anil, J. Lucas, and R. Grosse, “Sorting Out Lipschitz Function Approximation,” in *Proc. of the ICML*. PMLR, 2019, pp. 291–301.
- [122] A. Buades, B. Coll, and J.-M. Morel, “A Non-Local Algorithm for Image Denoising,” in *Proc. of the CVPR*, 2005, pp. 60–65.
- [123] O. Köpüklü, M. Babae, S. Hörmann, and G. Rigoll, “Convolutional Neural Networks with Layer Reuse,” in *Proc. of the ICIP*, 2019.
- [124] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated Residual Transformations for Deep Neural Networks,” in *Proc. of the CVPR*, 2017, pp. 5987–5995.

REFERENCES

- [125] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. Molloy, and B. Edwards, “Adversarial Robustness Toolbox v0.10.0,” *CoRR*, vol. abs/1807.01069, 2018.
- [126] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, “Black-Box Adversarial Attacks with Limited Queries and Information,” in *Proc. of the ICML*, 2018.
- [127] S. Ding, Z. Chen, T. Zheng, and J. Luo, “RF-Net: a Unified Meta-Learning Framework for RF-Enabled One-Shot Human Activity Recognition,” in *Proc. of the SENSYS*, 2020, pp. 517–530.
- [128] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, “Towards 3D Human Pose Construction in the Wild: a Weakly-Supervised Approach,” in *Proc. of the MobiCom*, 2020.
- [129] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola, “Correcting Sample Selection Bias by Unlabeled Data,” *Proc. of the NIPS*, vol. 19, pp. 601–608, 2006.
- [130] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Büna, and M. Kawanabe, “Direct Importance Estimation for Covariate Shift Adaptation,” *Annals of the Institute of Statistical Mathematics*, vol. 60, no. 4, pp. 699–746, 2008.
- [131] H. Daumé III, “Frustratingly Easy Domain Adaptation,” in *Proc. of the ACL*, 2007, pp. 256–263.
- [132] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, “Unsupervised Visual Domain Adaptation Using Subspace Alignment,” in *Proc. of the ICCV*, 2013, pp. 2960–2967.
- [133] B. Sun, J. Feng, and K. Saenko, “Return of Frustratingly Easy Domain Adaptation,” in *Proc. of the AAAI*, 2016, pp. 2058–2065.
- [134] P. Luo, F. Zhuang, H. Xiong, Y. Xiong, and Q. He, “Transfer Learning from Multiple Source Domains via Consensus Regularization,” in *Proc. of the CIKM*, 2008, pp. 103–112.
- [135] L. Duan, D. Xu, and I. W.-H. Tsang, “Domain Adaptation from Multiple Sources: A Domain-Dependent Regularization Approach,” *IEEE Transactions on neural networks and learning systems*, vol. 23, no. 3, pp. 504–518, 2012.
- [136] T. Evgeniou and M. Pontil, “Regularized Multi-Task Learning,” in *Proc. of the SIGKDD*, 2004, pp. 109–117.
- [137] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How Transferable are Features in Deep Neural Networks?” in *Advances in neural information processing systems*, 2014, pp. 3320–3328.
- [138] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin, “How Transferable are Neural Networks in NLP Applications?” in *Proc. of the EMNLP*, 2016, pp. 479–489.
- [139] Y. Ganin and V. Lempitsky, “Unsupervised Domain Adaptation by Backpropagation,” in *Proc. of the ICML*. PMLR, 2015, pp. 1180–1189.
- [140] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-Adversarial Training of Neural Networks,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [141] 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.
- [142] Y. Zhang, H. Tang, K. Jia, and M. Tan, “Domain-Symmetric Networks for Adversarial Domain Adaptation,” in *Proc. of the CVPR*, 2019, pp. 5031–5040.
- [143] C. Finn, P. Abbeel, and S. Levine, “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks,” in *Proc. of the ICML*, 2017, pp. 1126–1135.
- [144] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, “A Simple Neural Attentive Meta-Learner,” in *Proc. of the ICLR*, 2018.
- [145] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, “Matching Networks for One Shot Learning,” in *Proc. of the NIPS*, 2016, pp. 3630–3638.

REFERENCES

- [146] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil, “Bilevel Programming for Hyperparameter Optimization and Meta-Learning,” in *Proc. of the ICML*, 2018, pp. 1568–1577.
- [147] J. Snell, K. Swersky, and R. Zemel, “Prototypical Networks for Few-Shot Learning,” in *Proc. of the NIPS*, 2017, pp. 4077–4087.
- [148] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi, “Meta-Learning with Differentiable Closed-Form Solvers,” in *Proc. of the ICLR*, 2019.
- [149] A. Nichol and J. Schulman, “Reptile: A Scalable Metalearning Algorithm,” *arXiv preprint arXiv:1803.02999*, vol. 2, p. 2, 2018.
- [150] M. A. Jamal and G.-J. Qi, “Task Agnostic Meta-Learning for Few-shot Learning,” in *Proc. of the 32nd IEEE CVPR*, 2019, pp. 11 719–11 727.
- [151] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proc. the ICLR*, 2015.
- [152] Z. Li, F. Zhou, F. Chen, and H. Li, “Meta-SGD: Learning to Learn Quickly for Few-shot Learning,” *arXiv preprint arXiv:1707.09835*, 2017.
- [153] A. Antoniou, H. Edwards, and A. Storkey, “How to Train Your MAML,” in *International Conference on Learning Representations*, 2018.
- [154] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, “Learning to Learn by Gradient Descent by Gradient Descent,” in *Advances in neural information processing systems*, 2016, pp. 3981–3989.
- [155] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese Neural Networks for One-Shot Image Recognition,” in *Proc. of the ICML Workshop*, vol. 2, 2015.
- [156] J.-H. Kim, J. Jun, and B.-T. Zhang, “Bilinear Attention Networks,” in *Proc. of the NIPS*, 2018, pp. 1564–1574.
- [157] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proc. of the CVPR*, 2016, pp. 770–778.
- [158] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Proc. of the 32nd NIPS*, 2019, pp. 8024–8035.
- [159] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *Proc. of the ICLR*, 2015.
- [160] X. Zhang, X. Zhou, M. Lin, and J. Sun, “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices,” in *Proc. of the CVPR*, 2018, pp. 6848–6856.
- [161] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, “Benchmark Analysis of Representative Deep Neural Network Architectures,” *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.
- [162] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, “A Closer Look at Few-shot Classification,” in *Proc. of the ICLR*, 2019, pp. 1–16.
- [163] L. Liu, T. Zhou, G. Long, J. Jiang, and C. Zhang, “Many-class few-shot learning on multi-granularity class hierarchy,” *IEEE Transactions on Knowledge and Data Engineering*, 2020.
- [164] M. Goldblum, L. Fowl, and T. Goldstein, “Adversarially Robust Few-Shot Learning: A Meta-Learning Approach,” *arXiv*, p. Proc. of the NIPS, 2020.
- [165] C. Yin, J. Tang, Z. Xu, and Y. Wang, “Adversarial Meta-Learning,” *arXiv preprint arXiv:1806.03316*, 2018.
- [166] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, “Robustness may be at Odds with Accuracy,” in *Proc. of the ICLR*, 2019.

REFERENCES

- [167] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv preprint arXiv:1704.04861*, 2017.
- [168] A. Krizhevsky *et al.*, “Learning Multiple Layers of Features From Tiny Images,” *Tech Report*, 2009.
- [169] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, “Caltech-UCSD Birds 200,” California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.
- [170] C. E. Duchon, “Lanczos Filtering in One and Two Dimensions,” *Journal of applied meteorology*, vol. 18, no. 8, pp. 1016–1022, 1979.