ROBUST COMPUTER VISION AGAINST ADVERSARIAL EXAMPLES AND DOMAIN SHIFTS

by

Shao-Yuan Lo

A dissertation submitted to Johns Hopkins University
in conformity with the requirements for the degree of
Doctor of Philosophy

Baltimore, Maryland
March, 2023

Abstract

Recent advances in deep learning have achieved remarkable success in various computer vision problems. Driven by progressive computing resources and a vast amount of data, deep learning technology is reshaping human life. However, Deep Neural Networks (DNNs) have been shown vulnerable to adversarial examples, in which carefully crafted perturbations can easily fool DNNs into making wrong predictions. On the other hand, DNNs have poor generalization to domain shifts, as they suffer from performance degradation when encountering data from new visual distributions. We view these issues from the perspective of robustness. More precisely, existing deep learning technology is not reliable enough for many scenarios, where adversarial examples and domain shifts are among the most critical. The lack of reliability inevitably limits DNNs from being deployed in more important computer vision applications, such as self-driving vehicles and medical instruments that have major safety concerns.

To overcome these challenges, we focus on investigating and addressing the robustness of deep learning-based computer vision approaches. The first part of this thesis attempts to robustify computer vision models against adversarial examples. We dive into such adversarial robustness from four aspects: novel attacks for strengthening benchmarks, empirical defenses validated by a third-party evaluator, generalizable defenses that can defend against multiple and unforeseen attacks, and

defenses specifically designed for less explored tasks. The second part of this thesis improves the robustness against domain shifts via domain adaptation. We dive into two important domain adaptation settings: unsupervised domain adaptation, which is the most common, and source-free domain adaptation, which is more practical in real-world scenarios. The last part explores the intersection of adversarial robustness and domain adaptation fields to provide new insights for robust DNNs. We study two directions: adversarial defense for domain adaptation and adversarial defense via domain adaptations. This dissertation aims at more robust, reliable, and trustworthy computer vision.

Thesis Readers

Prof. Vishal M. Patel (Primary Advisor)
Associate Professor
Department of Electrical and Computer Engineering
Johns Hopkins University

Prof. Rama Chellappa
Bloomberg Distinguished Professor
Department of Electrical and Computer Engineering
Johns Hopkins University

Prof. Alan Yuille
Bloomberg Distinguished Professor
Department of Computer Science
Johns Hopkins University

Prof. Jesus Villalba
Assistant Research Professor
Department of Electrical and Computer Engineering
Johns Hopkins University

Acknowledgments

I thank Professor Vishal M. Patel, my Ph.D. advisor. I always remember the first email I sent to Prof. Patel in January 2019, which expressed my interest in joining his lab. He replied: "Your background interests me. We'll be in touch." Two weeks later, I received an offer letter from JHU. I cannot tell how excited I was. We made the first phone call in May to discuss research ideas, where I was at the crowded Taipei Station. In August 2019, I came to Baltimore. Prof. Patel involved me in DARPA's newly-initiated big project GARD to explore adversarial robustness. This project and research direction have been the primary path throughout my Ph.D. journey. Under his supervision, I underwent solid training for doing timely and advanced research, and have delivered a series of publications. He brought me to realize my dream of obtaining a Ph.D. in the U.S. Without his support, I would not have made current achievements. Furthermore, Prof. Patel's personality inspired me. He always encourages me whenever I presented externally or sometimes felt frustrated with paper rejections. I also learned a lot from how he runs a research lab and his aspiration. Joining VIU Lab is one of the best decisions I have made in my life. These will definitely motivate my career. Besides, I would like to thank Professors Alan Yuille, Rama Chellappa and Jesus Villalba for serving on my thesis committee, and Professors Yinzhi Cao and Jeremias Sulam for serving on my GBO committee, and Professors

Najim Dehak and Trac D. Tran for serving on my DQE committee, and for their valuable feedback.

I am grateful for the support of all my Ph.D. colleagues and friends at VIU Lab, especially Poojan Oza and Jeya Maria Jose. I am fortunate to have collaborated with and been mentored by Poojan. He taught me to better make plans for new projects, come up with new ideas, write rebuttals, interact with managers, etc. I presented three out of all my nine Ph.D. research works in my thesis defense, where two were worked with him. I will miss the time we went for lunch and movies during my internship in Seattle. To Jose, who joined the lab at the same time as me. We have experienced our lab's changes before, during, and after the COVID-19 pandemic throughout our 4-year Ph.D. journey. I am also thankful to Dana Walter-Shock and Belinda Blinkoff, the administrators of the ECE Dept., for their support from my admission to graduation.

I am deeply grateful to all the friends I have made in JHU Taiwanese Student Association from my first to last years. I had served as the Vice President of TSA and have spent most of my leisure time with these amazing friends. Some special words of gratitude go to the Teahouse Gang, who have been with me since my first year in Baltimore. They have seen my growth in certain fields, and we have spent time together in Baltimore, NYC, NJ, PA, Seattle, SF, and my hometown Zhudong Township. We have always had valuable conversations. To my Huaihuai buddy, a very supportive friend, we have hung out countless times and shared our valuable experiences. And to the 905 Gang, who were among the most important during my last year. I will miss the hot pot parties, fortune-telling, and weekly coffee and study meetups. My Ph.D. life would not be colorful without these TSA friends.

I extend my appreciation to all the friends I made in the ECE Dept., particularly

the Yihao Academic Gang, with whom I had almost taken all classes together and dined frequently. I always remember when I have not had a car in my first year, they drove me to grocery stores and to car dealers to buy my first car during the pandemic. Even after they graduated, some of us have reunited twice in Seattle during my internship, and some of us have had research collaborations. Besides, I am thankful to my friends at RUF International House, with whom I enjoyed sharing cross-strait cultures. A special thanks to the 30W Gang and Terry, with whom I attended the weekly English groups, celebrated Christmas and Lunar New Year, and experienced indoor gun shootings. I had never expected that I would make so many friends in Baltimore. During my Ph.D. study, I am fortunate to have interned at Amazon Astro and Amazon JWO. A shout-out to Doctors Jim Thomas and Alejandro Galindo, the managers of my two internships respectively. They gave me industry experience and positive evaluations with possible return opportunities. Many thanks to the friends who I made or had been with me in Seattle. I will miss the two incredible summers in 2021 and 2022.

Finally, I would like to thank my family. I still remember the deep concern my parents expressed when I received an offer from JHU in January 2019. I can imagine how difficult it must be for any parent to let their child independently go study abroad across half of Earth, most importantly such an extremely unusual child. I want to express my great gratitude to my parents for letting me be free to pursue my dreams and life journey. Thanks to them for their love and support throughout my life. Moreover, many thanks to my brother and all my family members, who have cheered me on all my achievements. My family is the best family.

To my grandfather Kan Lo, my grandmother Hsiu-Lien Lo-Tseng,
my father Wen-Shun Lo, my mother Ya-Wen Yang, and my brother Shao-Wei Lo
for their unwavering love granted over years.

To all the people who have supported me in my life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

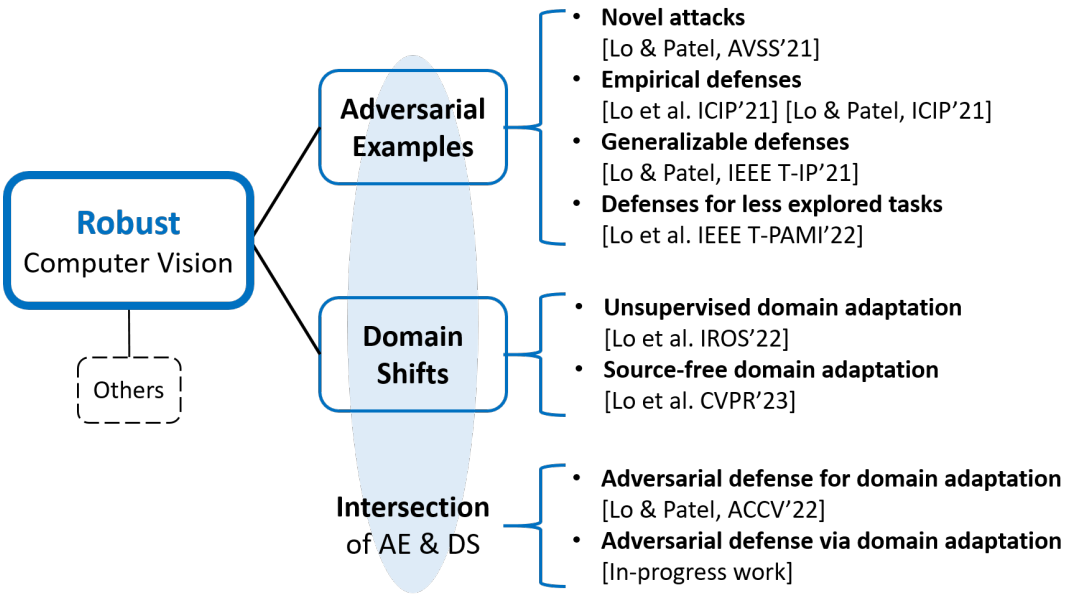

Figure 1.1: Overview of this dissertation.

This dissertation focuses on robust computer vision. Deep learning technology has achieved remarkable success in various computer vision applications [9, 25, 37, 39, 44, 69, 180], but it is not reliable enough for many real-world scenarios [23, 26, 196]. The robustness issue of deep learning has many aspects, where adversarial examples and domain shifts are among the most critical. In this thesis, we first attempt to robustify computer vision models against adversarial examples. Next, we improve the robustness against domain shifts by domain adaptation. Finally, we explore the intersection of adversarial robustness and domain adaptation. Figure 1.1 presents an overview of this dissertation.

Adversarial examples. Adversarial examples $\mathbf{x}_{adv}$ are generated by adding carefully-crafted perturbations δ_{adv} to the input data $\mathbf{x}$:

$$\mathbf{x}_{adv} = \mathbf{x} + \delta_{adv}. \tag{1.1}$$

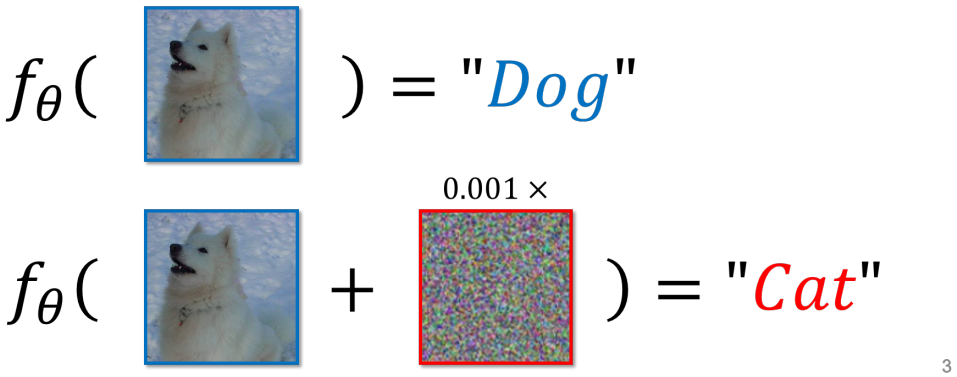

Figure 1.2: Illustration of adversarial examples. Adding carefully-crafted perturbations with a very small magnitude to the input image can fool a well-trained model. The example dog image is from ImageNet [35].

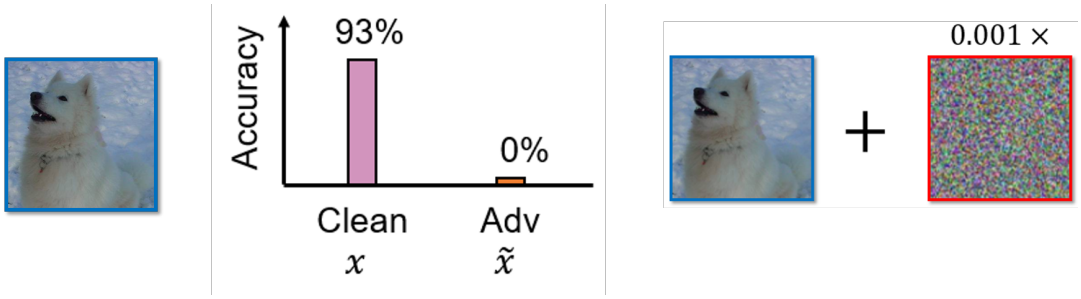

Figure 1.3: An example result of adversarial examples. Network: ResNet-50 [69]. Dataset: CIFAR-10 [103].

The adversarial examples can fool well-trained Deep Neural Networks (DNNs) to make wrong predictions:

$$f_\theta(\mathbf{x}_{adv}) \neq y, \tag{1.2}$$

where f_θ is a target model with well-trained parameters, and y deontes the ground-truth label. Such perturbations are very small and usually imperceptible or insusceptible to human eyes. Figure 1.2 provides an illustration. A well-trained DNN can correctly recognize the dog image. However, if we add carefully-crafted perturbations with a very small magnitude to the input image, the DNN misclassifies the dog image as

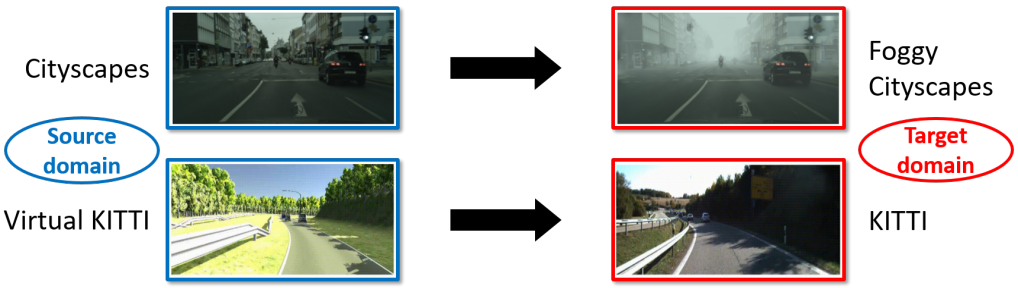

Figure 1.4: Illustration of domain shifts. The source domain (e.g., Cityscapes [32], Virtual KITTI [51]) and the target domain (e.g., Foggy Cityscapes [184], KITTI [55]) have similar contents but different appearances.

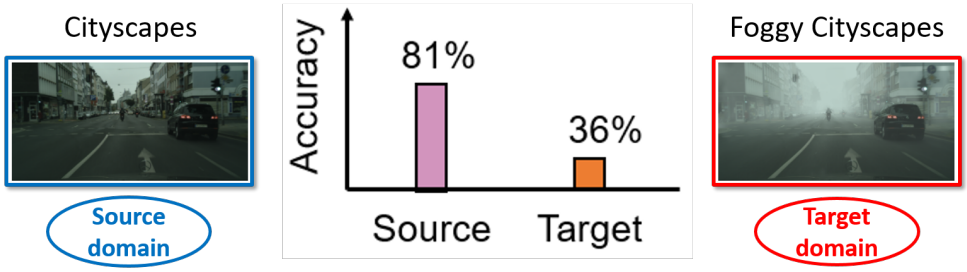

Figure 1.5: An example result of domain shifts. Network: DeepLabv2 [25]. Benchmark: Cityscapes [32] to Foggy Cityscapes [184].

a cat. It has been known that DNNs are vulnerable to adversarial examples [13, 60, 201]. Figure 1.3 shows an example result. A well-trained ResNet-50 [69] network can achieve 93% accuracy on the CIFAR-10 [103] dataset. However, if we add adversarial perturbations to input images, the accuracy drops to 0%. Hence, proposing robust models against adversarial examples is needed.

Domain shifts. Consider a scenario in which the training (source) data and test (target) data are from different domains (i.e., datasets). In this scenario, accuracy would drop on target data due to the domain shift problem. To address this problem, domain adaptation is introduced. The classic domain adaptation is formulated as that: Given

a labeled source dataset and an unlabeled target dataset, learn a model for the target domain. Figure 1.4 provides an illustration. Given a road scene semantic segmentation model, DeepLabv2 [25] that is trained on images with sunny weather (e.g., Cityscapes [32]), its performance would be poor on images with foggy weather (e.g., Foggy Cityscapes [184]). In addition, if a model is trained on synthetic virtual data (e.g., Virtual KITTI [51]), its performance would be poor on realistic data (e.g., KITTI [55]). Figure 1.5 shows an example result. Given a model trained on the Cityscapes dataset, it can achieve 81% mean of Intersection-over-Union (mIoU) accuracy on the same dataset. However, if we test the model on the Foggy Cityscapes dataset, its mIoU accuracy drops to 36%. Therefore, developing robust models against domain shifts is important.

Overview. For adversarial examples, we study several aspects, including novel attacks, empirical defenses, generalizable defenses, and defenses for less explored tasks. For domain shifts, we study Unsupervised Domain Adaptation (UDA) and Source-Free Domain Adaptation (SFDA). For their intersection, we explore adversarial defense for domain adaptation and adversarial defense via domain adaptation. The following sections elaborate background and the contributions of this thesis.

1.1 Robust Computer Vision Against Adversarial Examples

Existing studies reveal that DNN-based computer vision models are vulnerable to adversarial examples [13, 201], and many adversarial attack algorithms have been proposed. Fast Gradient Sign Method (FGSM) [60] leverages the sign of gradients to produce adversarial examples. Projected Gradient Descent (PGD) [150] extends

FGSM from single iteration gradient descent to an iterative version with a random start. MI-FGSM [38] generates more transferable adversarial attacks via a momentum mechanism. We notice that most attack algorithms are additive attacks, in which perturbations are added to input data.

Various adversarial defenses have also been introduced. Earlier approaches rely on image transformations as pre-processing, but they fail to defend against white-box attacks [5, 18]. In contrast, Adversarial Training (AT) [60] has been considered the most effective strategy. AT trains a model on adversarial examples generated on the fly according to the model's current parameters. PGD-AT [150] formulates AT in a min-max optimization framework and trains a model with only adversarial examples. Feature Denoising (FD) [239] appends the feature denoising blocks to a model to remove adversarial perturbations in the feature domain. TRADES [256] minimizes a regularized surrogate loss to obtain a better trade-off between adversarial robustness and clean data accuracy. However, most AT defenses are robust to only a single perturbation type. Recent works like AVG [205], MAX [205] and Multi Steepest Descent (MSD) [153] can defend against different L_p-norm perturbations. Still, they do not consider physically realizable attacks (e.g., patch attacks [15, 46, 203, 231]) and unforeseen attacks. In addition, most adversarial robustness studies focus on image classification. Many other computer vision tasks, such as action recognition and novelty detection, remain less explored.

1.1.1 Contributions

In Part I, we dive into adversarial robustness from four aspects (see Figure 1.1). First, diverse attack algorithms are critical to evaluate a model's robustness, so we

propose a novel attack to strengthen existing benchmarks. Second, we propose new empirical adversarial defenses whose effectiveness has been validated by a third-party evaluator. Third, we propose a generalizable defense that can defend against multiple and unforeseen attacks. Finally, we propose a defense specifically designed for novelty detection, which is less explored.

Novel attacks. In Chapter 3, we notice that most attack algorithms are additive attacks, in which perturbations are added to input data (see Eq. (1.1)). To this end, we propose MultAdv [141], a novel multiplicative attack that imposes perturbation by multiplication instead of addition. MultAdv has different noise distributions to the additive counterparts and thus challenges the defense methods tailored to resisting additive adversarial attacks. For example, the FD defense has 42.7% accuracy under the additive PGD attack on UCF-101 [198], an action recognition dataset. Nevertheless, its accuracy decreases to 31.5% under our multiplicative MultAdv attack. MultAdv increases the attack diversity of existing robustness benchmarks and motivates researchers to develop defenses that can resist more attack types.

Empirical defenses. The empirical defense is one of the mainstreams of robustness research. It can protect models to the largest extent. AT is considered the most effective strategy. On top of AT, Chapter 4 proposes OUDefend [143], which is one of the first defenses for action recognition. It learns both Over-and-Under complete representations. Under-complete representations have large receptive fields to collect global information but overlook local details, while over-complete representations have opposite properties. OUDefend is designed to balance both local and global features by learning those two representations. Hence, it can capture perturbation patterns more precisely and further remove them in the feature space. Experiments

show that OUDefend (49.5% accuracy) outperforms both PGD-AT (48.0%) and FD (42.2%) on UCF-101.

Furthermore, Chapter 5 proposes the Halftoning defense [140], a novel image transformation-based defense using error diffusion halftoning. Error diffusion halftoning projects an image into a 1-bit space and diffuses quantization error to neighboring pixels. This process can remove adversarial perturbations from a given image while maintaining acceptable image quality in the meantime in favor of recognition. Although most image transformation-based defenses are ineffective under white-box attacks, Halftoning can still improve robustness. We submit this work to the DARPA GARD [40] project with a third-party evaluator. Their results show that when combing AT, Halftoning achieves 92% accuracy under both PGD and Masked PGD attacks on UCF-101, which is the best method for action recognition over all the submissions of this project.

Generalizable defenses. Most AT-based defenses are limited to a specific type of adversarial perturbations. They often fail to offer resistance to multiple attack types simultaneously, i.e., they lack multi-perturbation robustness. To address this, Chapter 6 proposes a new generalizable defense, MultiBN [139], based on a multiple Batch Normalization (BN) structure and a BN selection module. It performs AT on multiple adversarial perturbation types using multiple independent BN layers with a learning-based BN selection module. Compared to related works AVG (17.3% union accuracy on the UCF-101 dataset), MAX (5.5%) and MSD (0.7%), MultiBN (34.8%) exhibits much stronger multi-perturbation robustness against different and even unforeseen adversarial perturbation types, ranging from L_p-norm and physically realizable attacks.

Defenses for less explored tasks. Most adversarial robustness studies focus on the image classification task. Many other computer vision tasks remain less explored, and novelty detection is one of them. A novelty detector is trained with examples of a particular class and is tasked with identifying whether a query example belongs to the same known class. In Chapter 7, we propose Principal Latent Space (PrincipaLS) [138], a novel defense that learns the incrementally-trained cascade principal components in the latent space to robustify novelty detectors. It can purify latent space against adversarial examples and constrain latent space to exclusively model the known class distribution. We conduct extensive experiments on 8 attacks, 5 datasets and 7 novelty detectors, showing that PrincipaLS consistently enhances the robustness of novelty detection models.

1.2 Robust Computer Vision Against Domain Shifts

DNN-based computer vision models suffer from performance degradation when encountering data from new visual distributions. This is known as the domain shift problem. UDA tackles domain shifts by aligning the representations of the source and target domains [52]. For example, AdaDepth [109] employs adversarial learning at both feature and output spaces to align the distributions between the two domains. T2Net [260] transfers source images to the target style to train a model. GASDA [259] uses bidirectional style transfer to learn the mapping between two domains. Nevertheless, they either have suboptimal domain alignment or high computational complexity during inference.

In real-world applications, source data are often restricted because of concerns about data privacy, commercial proprietary, transmission efficiency, etc. Under this

scenario, UDA is not applicable since it relies on accessing the source data to reduce the domain gap. Recently, SFDA has been introduced to address this issue [30, 121, 125]. It adapts a source-trained model to the target domain without requiring access to source data. SFDA-SS [132] develops a data-free knowledge distillation strategy for adaptation. UR [197] reduces the uncertainty of target data predictions. HCL [77] presents historical contrastive learning, which leverages the historical source hypothesis to compensate for the absence of source data. However, SFDA is still unexplored in videos, where existing approaches do not consider temporal information.

1.2.1 Contributions

In Part II, we dive into two important domain adaptation settings (see Figure 1.1). First, we study the most common UDA setting. Next, given that UDA is impractical in many real-world scenarios, we further study the more challenging SFDA setting.

Unsupervised domain adaptation. UDA tackles the domain shift problem by aligning the representations of the source and target domains. Chapter 8 proposes a new UDA method, referred to as Learning Feature Decomposition for Adaptation (LFDA) [144], for the Monocular Depth Estimation (MDE) task. Domain adaptive MDE is less explored, especially compared to image classification and semantic segmentation. LFDA learns to decompose the feature space into content and style components, where it only attempts to align the content component since it has a smaller domain gap. Moreover, LFDA uses separate feature distribution estimations to further bridge the domain gap. Experiments show that achieves higher accuracy and 64% faster inference speed than GASDA on the standard Virtual KITTI-to-KITTI [51, 55] adaptation benchmark.

Source-free domain adaptation. SFDA aims to adapt a source-trained model to the target domain without requiring access to source data, which is more feasible for real-world applications. In Chapter 9, we notice that SFDA remains unexplored in videos, so we explore a spatio-temporal extension of SFDA for Video Semantic Segmentation (VSS). We propose a novel method, namely Spatio-Temporal Pixel-Level contrastive learning (STPL) [137], which takes full advantage of spatio-temporal information for video adaptation. Specifically, STPL explicitly learns semantic correlations among pixels in the spatio-temporal space, providing strong self-supervision for adaptation to the unlabeled target domain. Experiments on VSS benchmarks show the superiority of STPL over image-based SFDA and even UDA approaches relying on source data.

1.3 Intersection of Adversarial Robustness and Domain Adaptation

Although adversarial robustness and domain adaptation are traditionally treated as two individual research fields, they actually have similarities and their intersection is worth exploring. Both of them raise the reliability issue of DNNs, which is one of the main weaknesses of modern deep learning technology. Specifically, DNNs can achieve great accuracy on clean samples, yet they could not resist intentional worst-case perturbations. Similarly, DNNs have performed excellently on the data from the same distribution as their training data, but their performance degrades when encountering data from new distributions. These points out that DNNs work well under only specific circumstances, which limits them from being deployed in broader applications.

Several recent studies investigate the adversarial robustness of domain adaptation.

RFA [6] leverages multiple external adversarially pre-trained models as teacher models to distill robustness knowledge. ASSUDA [245] employs an external pre-trained UDA model that generates pseudo labels for unlabeled target data to do AT. Nevertheless, RFA's performance is highly sensitive to the teacher models' setup, and ASSUDA considers only the weak black-box attacks instead of the strong white-box attacks.

On the other hand, a few works investigate adversarial effects via domain adaptation techniques. For instance, AdvProp [236] uses an auxiliary BN branch to learn clean and adversarial feature distributions separately, which improves image recognition. This idea is originally from the domain adaptation field [22]. DRRDN [247] disentangles clean and adversarial distributions to improve robustness. This is inspired by domain adaptation as well [21]. Enhancing adversarial robustness from the perspective of domain adaptation is a promising direction and still an open problem.

1.3.1 Contributions

In Part III, we investigate the intersection of adversarial robustness and domain adaptation from two aspects (see Figure 1.1): adversarial defense for domain adaptation, where we propose a defense for UDA; and adversarial defense via domain adaptation, where we propose a defense via domain adaptation techniques.

Adversarial defense for domain adaptation. Little focus is devoted to improving UDA's adversarial robustness. Despite AT's success, it requires ground-truth labels to generate adversarial examples and train models, which limits its effectiveness in the unlabeled target domain. To this end, Chapter 10 provides a systematic study into multiple AT variants that can potentially be applied to UDA. Based on that, we propose a novel Adversarially Robust Training method for UDA, referred to as ARTUDA

[142]. Compared to RFA, which achieves 34.1% robust accuracy on the VisDA-2017 [167] dataset, ARTUDA improves the UDA model's robust accuracy to 40.7%.

Adversarial defense via domain adaptation. Despite AT's success, its training cost is extremely high. In Chapter 11, we attempt to develop a domain adaptation-motivated adversarial defense to get rid of the expensive AT. Inspired by Test-Time Adaptation (TTA) ideas [124, 212, 223], we propose Adaptive BN Network (ABNN), a non-AT defense method. ABNN employs a pre-trained substitute model to generate clean BN statistics and send them to the target model. The target model is exclusively trained on clean images and learns to align the substitute model's BN statistics. Results show that ABNN can improve robust accuracy without using AT, and it achieves higher clean accuracy than PGD-AT.

The main ideas of this dissertation are composed of the following publications:

1. **Shao-Yuan Lo** and Vishal M. Patel. "Multav: Multiplicative adversarial videos". In: *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2021. – [Chapter 3]

2. **Shao-Yuan Lo**, Jeya Maria Jose Valanarasu, and Vishal M. Patel. "Overcomplete representations against adversarial videos". In: *IEEE International Conference on Image Processing (ICIP)*. 2021. – [Chapter 4]

3. **Shao-Yuan Lo** and Vishal M. Patel. "Error diffusion halftoning against adversarial examples". In: *IEEE International Conference on Image Processing (ICIP)*. 2021. – [Chapter 5]

4. **Shao-Yuan Lo** and Vishal M. Patel. "Defending against multiple and unforeseen adversarial videos". In: *IEEE Transactions on Image Processing (T-IP)*. 2021. – [Chapter 6]

5. **Shao-Yuan Lo**, Poojan Oza, and Vishal M. Patel. "Adversarially robust one-class novelty detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*. 2022. – [Chapter 7]

6. **Shao-Yuan Lo**, Wei Wang, Jim Thomas, Jingjing Zheng, Vishal M. Patel, and Cheng-Hao Kuo. "Learning feature decomposition for domain adaptive monocular depth estimation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022. – [Chapter 8]

7. **Shao-Yuan Lo**, Poojan Oza, Sumanth Chennupati, Alejandro Galindo, and Vishal M. Patel. "Spatio-temporal pixel-level contrastive learning-based source-free domain adaptation for video semantic segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023. – [Chapter 9]

8. **Shao-Yuan Lo** and Vishal M. Patel. "Exploring adversarially robust training for unsupervised domain adaptation". In: *Asian Conference on Computer Vision (ACCV)*. 2022. – [Chapter 10]

Chapter 2

Background

In this chapter, we review the background knowledge and related works of this dissertation that will be frequently discussed in the following chapters. Our review covers adversarial attacks, adversarial defenses, adversarial videos, and domain adaptation. Lastly, we provide a list of abbreviations commonly used in this dissertation.

2.1 Adversarial Attacks

Recent advances in deep learning have led DNNs to perform outstandingly well in many computer vision problems [7, 25, 68, 69], including tasks such as video recognition [19, 37, 65]. However, researchers have shown that DNNs are easily misled when presented by adversarial examples [60, 201]. The adversarial examples are intentionally constructed or collected by humans to fool DNNs into making wrong predictions [13, 73], which pose serious security threats.

Most current studies construct the adversarial examples by adding intentionally worst-case perturbations to input data [15, 60, 150, 201, 253]. Szegedy et al. [201] first showed that carefully crafted perturbations can fool DNNs. Goodfellow et al. [60] introduced the Fast Gradient Sign Method (FGSM), which leverages the sign of gradients to produce adversarial examples. Projected Gradient Descent (PGD) [150] extends FGSM from single iteration gradient descent to an iterative version. Momentum Iterative FGSM (MI-FGSM) [38] generates more transferable adversarial attacks by a momentum mechanism. These attacks are categorized into the L_p*-norm attack*, as they are bounded by L_p space. Another type of attack called *physically realizable attack* is also developed. Different from L_p-norm attacks, physically realizable attacks' perturbations are printable and thus can be deployed in the physical world. For example, Rectangular Occlusion Attack (ROA) [231] puts rectangular

adversarial patches on input images, and Adversarial Framing (AF) [253] produces framings whose perturbations are on the border of images. In the following, we elaborate on these two main adversarial attack categories. We also propose a new L_p-norm attack, namely Salt-and-Pepper Attack (SPA).

2.1.1 Lp-norm Attacks

PGD. PGD [150] is the most widely-used attack approach. Given a data sample x, ground-truth label y, model parameters θ, and training loss $\mathcal{L}$, PGD generates adversarial example $\tilde{x}$ in an iterative way:

$$x^{t+1} = Proj^{L_p}_{x,\epsilon}\{x^t + \alpha \cdot sign(\bigtriangledown_{x^t}\mathcal{L}(x^t, y; \theta))\}, \tag{2.1}$$

where α is step size, $t \in [0, t_{max} - 1]$ denotes attack iteration and thus $x = x^0$ and $\tilde{x} = x^{t_{max}}$. $Proj^{L_p}_{x,\epsilon}\{\cdot\}$ projects its element into a L_p bound with perturbation size ϵ such that $\| x^{t+1} - x \|_p \leq \epsilon$. PGD is a multi-step with a random start variant of FGSM. It has become one of the most important benchmarks in current adversarial robustness research.

MI-FGSM. MI-FGSM is based on FGSM/PGD and adopts a momentum mechanism in the attack formulation:

$$g^{t+1} = \mu \cdot g^t + \frac{\bigtriangledown_{x^t}\mathcal{L}(x^t, y; \theta))}{\| \bigtriangledown_{x^t}\mathcal{L}(x^t, y; \theta) \|_1}, \tag{2.2}$$

where g^t gathers the gradients of the first t iterations with a decay factor μ. Then,

$$x^{t+1} = Proj^{L_p}_{x,\epsilon}\{x^t + \alpha \cdot sign(g^{t+1})\}. \tag{2.3}$$

This formulation achieves better attack transferability due to the momentum.

SPA. In this chapter, we propose SPA, a new L_p-norm attack inspired by the one-pixel attack [200]. For computation saving, instead of using differential evolution, SPA randomly selects a pre-defined number of pixels on an image, then applies PGD on these pixels. We consider SPA as a kind of L_0-norm attack because the number of adversarial pixels is bounded. The generated perturbations look like salt-and-pepper noise. Proposing SPA allows us to better benchmark adversarial defense approaches.

2.1.2 Physically Realizable Attacks

The *physical attack* is a category of adversarial attacks that is implemented in the physical space. The physically realizable attack refers to a digital representation of the physical attack. Such attacks fool DNNs by modifying physical objects being photographed. Sharif et al. [191] generated printable perturbations inside eyeglass frames to attack face recognition systems. Brown et al. [15] created an adversarial patch that can be put next to a real-world object, making that object misclassified. Thys et al. [203] further extended the adversarial patch to fool human detectors.

ROA performs L_∞-norm PGD inside a fixed size and fixed location rectangle on an image. The size is pre-defined, and the location is searched with respect to the highest loss that it can cause. The perturbations look like a rectangular adversarial sticker on an image. AF adds adversarial perturbations on the border of an image, while the remaining pixels are kept unchanged. It first fixes the framing size and location, then performs L_∞-norm PGD inside the framing. ROA and AF can be formulated as follows:

$$x^{t+1} = Proj^{L_p}_{x,\epsilon}\{x^t + m \cdot \alpha \cdot sign(\bigtriangledown_{x^t}\mathcal{L}(x^t, y; \theta))\}, \tag{2.4}$$

where $m \in \{0, 1\}$ is a binary mask of ROA or AF. Let p be a pixel index of m. For

ROA, if p is inside the rectangular with a size of s_{ROA} within m, $m_p = 1$; otherwise, $m_p = 0$. Similarly, for AF, if p is on the border of m within a framing width s_{AF}, $m_p = 1$; otherwise, $m_p = 0$.

2.2 Adversarial Defenses

To defend against adversarial attacks, various defense approaches have also been proposed in the literature [1, 60, 64, 91, 97, 100, 110, 128, 150, 173, 188, 238, 239, 256]. Earlier attempts focus on detecting adversarial examples [71, 88, 123]. However, detection is inherently weaker than defense in terms of resisting adversarial attacks. Several defenses based on image transformation are introduced [11, 64, 244], but they have been proven not robust against white-box attacks [5, 18].

Currently, Adversarial Training (AT) based defenses [60, 97, 150, 256] have been considered the most effective, especially under the white-box setting. The core idea is to train a model on adversarial examples that are generated on-the-fly according to the model's current parameters. Madry et al. [150] formulated AT in a min-max optimization framework (PGD-AT), and this has been widely used as a benchmark. Xie et al. [239] includes the Feature Denoising (FD) block in networks to remove adversarial perturbations in the feature domain. Hendrycks et al. [72] added an auxiliary rotation prediction task [56] to improve PGD-AT (i.e., RotNet-AT). TRADES [256] minimizes a regularized surrogate loss to obtain a better trade-off between robustness and performance, where both clean data and adversarial examples are used for training. Smooth Adversarial Training (SAT) [237] uses smooth approximations of the ReLU [160] activation to enhance PGD-AT. Self-supervised Online Adversarial Purification (SOAP) [193] employs self-supervised signals to

purify adversarial examples during inference. In the following, we elaborate on the image transformation-based defenses and AT.

2.2.1 Image Transformation-based Methods

Image transformation-based defenses deploy an image transformation at the pre-processing stage before inference to protect DNNs from adversarial effects. The intuition is that such image pre-processing could filter out adversarial perturbations, allowing one to feed an adversary-free image to a classifier. Specifically, consider a target classifier C, an adversarial example x_{adv} and its ground-truth label y, where $C(x_{adv}) \neq y$. The idea is to find a transformation T such that $C(T(x_{adv})) = y$. Many types of transformations have been adopted in the context of adversarial defense. Bit-depth reduction quantizes pixel values to invalidate adversarial variations in an image [64, 244]. JPEG compression performs quantization in the frequency domain to remove perturbations [34, 42, 91]. Image denoising operations such as mean filter, median filter and non-local means [17] have been used as defenses as well [123, 239]. Several approaches use a DNN-based denoiser as the image transformation. For instance, High-level representation Guided Denoiser (HGD) [126] adopts a U-Net-based [177] denoiser with high-level feature guidance against adversaries at the pixel level. ComDefend [91] mitigates adversarial effects through an image compression network-based denoiser.

However, almost all of these attempts have been defeated under the white-box threat model. Most of their effectiveness is actually caused by obfuscated gradients, which gives a false sense of robustness. Hence, if attackers are aware of the presence of the defense, they are able to incorporate the defense into adversary search and

combat obfuscated gradients. Raff et al. [173] tried to stochastically combine a lot of image transformations to defend against adaptive attacks in the white-box setting, but failed to maintain the performance on clean images. Their clean data performance is largely sacrificed owing to the multiple transformations. Therefore, this defense stream is declining.

2.2.2 Adversarial Training

AT is proven to provide strong robustness, especially against challenging white-box attacks. It has been used as a foundation for more advanced defense techniques. Goodfellow et al. [60] first proposed this strategy. They trained DNNs with both clean and adversarial data to improve adversarial robustness. Currently, PGD-AT is one of the most commonly used AT algorithms. Let us recall the objective function for training a DNN model:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\mathcal{L}(x, y; \theta)\right], \tag{2.5}$$

where x is a clean training sample with ground-truth label y in the training set $\mathbb{D}$, θ is model parameters, and $\mathcal{L}$ denotes the training loss. PGD-AT formulates AT as a min-max optimization problem and trains models exclusively with adversarial data:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\max_{\delta\in\mathbb{S}} \mathcal{L}(x+\delta, y; \theta)\right], \tag{2.6}$$

where δ denotes an adversarial perturbation that is within a bounded perturbation set $\mathbb{S}$.

2.3 Adversarial Videos

Most existing literature on adversarial attacks and defenses focuses on static images. There are only a few studies presenting attack techniques for videos. Wei et al. [228] was the first to explore adversarial examples in videos. They found that perturbations propagate through video frames in the video classifiers [37] that are based on a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN). According to this observation, they proposed a temporally sparse attack. Li et al. [122] produced adversarial videos by a generative model. Jiang et al. [93] developed V-BAD against video recognition networks in the black-box setting. Pony et al. [172] proposed a spatial patternless adversarial video, in which the perturbation is a constant offset applied to the entire video frame. Thus, the attack is constructed using temporal information exclusively without using any spatial information.

Similarly, not many defense techniques against adversarial videos have been presented. Xiao et al. [233] proposed AdvIT based on temporal consistency to detect adversarial frames within a video. However, their approach only detects whether a video has been attacked or not. It does not provide a defense mechanism against the attacked videos. Jia et al. [90] presented a similar detector along with a temporal defense and a spatial defense. The temporal defense reconstructs perturbed frames with adjacent clean frames. The spatial defense uses and denoises the reconstructed frames to mitigate the effect of adversarial perturbations. However, their approach is only evaluated on the black-box attack setting. It is not clear how well their defense works on white-box attacks.

2.4 Domain Adaptation

Recent advances in visual recognition have enjoyed remarkable success via deep supervised learning [69, 104, 252]. However, the *domain shift* problem is very common in real-world scenarios, i.e., the training (source) and test (target) data are from different domains and thus have different data characteristics. For example, a model trained with the images and annotations from a specific camera may not generalize well to the images from another camera with different camera settings (e.g., focal length, size of field view). Furthermore, it is costly and labor-intensive to collect the ground-truth labels of target data. Synthetic data and their annotations are easier to acquire, but models trained with such a synthetic dataset often suffer from severe accuracy degradation when tested on realistic data. These issues are known as the domain shift problem. To address domain shifts, domain adaptation is introduced. The goal is to transfer the knowledge from a labeled source dataset to an unlabeled target dataset. There have been many domain adaptation variants in the literature. This section reviews two common settings: Unsupervised Domain Adaptation (UDA) and Source-Free Domain Adaptation (SFDA).

2.4.1 Unsupervised Domain Adaptation

UDA is a classic and the most common domain adaptation setting. It considers the scenario that the source dataset contains data with labels, while label information is unavailable to the target dataset. The goal can be formulated as that given a labeled source dataset and an unlabeled target dataset, learn a model for the target domain. Existing approaches can be categorized into three main streams: Adversarial Learning (AL) [21, 41, 52, 53, 145, 146, 147, 208, 210, 220, 222], image-to-image translation

[2, 4, 29, 75, 148, 159, 171, 259, 260], and self-training [24, 78, 156, 183, 249, 263].

AL-based methods rely on feature distribution alignment, which aims to minimize distribution discrepancy between source and target domains to learn domain-invariant representations. Given a labeled source dataset $\mathbb{D}_s = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ and an unlabeled target dataset $\mathbb{D}_t = \{x_t^i\}_{i=1}^{n_t}$ with n_s and n_t number of samples, respectively, a typical AL-based model learns a feature extractor F and a classifier C on top of F. Given an input image x, we express its feature space representation as $F(x)$ and its output logits as $C(x)$, where we use $C(x)$ as a simplification of the formal expression $C(F(x))$. The objective function of an AL-based model can be written as:

$$\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{DA}\big(x_s, x_t\big), \tag{2.7}$$

where $\mathcal{L}_{CE}$ is the task training loss (e.g., cross-entropy loss), and $\mathcal{L}_{DA}$ is the domain adaptation loss defined by each AL-based method. One of the most common $\mathcal{L}_{DA}$ is the adversarial loss introduced by Domain-Adversarial Neural Networks (DANN) [52, 53], which is defined as:

$$\mathcal{L}_{DA}\big(x_s, x_t\big) = \mathbb{E}[logD(F(x_s))] + \mathbb{E}[1 - (logD(F(x_t)))], \tag{2.8}$$

where D is a domain discriminator used to encourage domain-invariant features. DANN proposes the first AL-based method. It uses a domain discriminator that distinguishes between source and target features, and the feature extractor is trained to fool it via the Generative Adversarial Network (GAN) [59] learning scheme. Adversarial Discriminative Domain Adaptation (ADDA) [210] combines DANN with discriminative feature learning. Conditional Domain Adversarial Network (CDAN) [146] extends DANN using a class-conditional adversarial game. Joint Adaptation

Network (JAN) [147] aligns the joint distributions of domain-specific layers between two domains.

Image-to-image translation-based methods perform source-to-target translation (i.e., the source data will have source data content with target data style) or vice versa to bridge the domain gap between the source and target domains. These methods are effective, but image translation [79, 83] itself is not an easy task. Images may not be perfectly translated to another domain or lead to contain distortion after translation. Self-training-based methods generate pseudo-labels for target data during the training process to guide the models in learning target domain knowledge.

2.4.2 Source-Free Domain Adaptation

UDA relies on the assumption that both source and target data are available during adaptation. In real-world scenarios, the access to source data is often restricted (e.g., data privacy, commercial proprietary) or infeasible (e.g., data transmission efficiency, portability). Hence, under these source data restrictive circumstances, UDA approaches are less practical.

To deal with these issues, SFDA, also referred to as Unsupervised Model Adaptation (UMA), has been introduced recently in the literature [30, 77, 107, 108, 121, 125, 132, 197, 199, 251]. SFDA aims to use a source-trained model (i.e., a model trained on labeled source data) and adapt it to an unlabeled target domain without requiring access to the source data. More precisely, under the SFDA formulation, given a source-trained model and an unlabeled target dataset, the goal is to transfer the learned source knowledge to the target domain. In addition to alleviating data privacy or proprietary concerns, SFDA makes data transmission much more efficient.

For example, a source-trained model ($\sim$ 0.1 - 1.0 GB) is usually much smaller than a source dataset ($\sim$ 10 - 100 GB). If one is adapting a model from a large-scale cloud center to a new edge device that has data with different domains, the source-trained model is far more portable and transmission-efficient than the source dataset.

2.5 List of Abbreviations

In the following, we list the abbreviations that will be frequently used in this thesis.

- DNN: Deep Neural Network
- FGSM: Fast Gradient Sign Method [60]
- PGD: Projected Gradient Descent [150]
- MI-FGSM: Momentum Iterative FGSM [38]
- ROA: Rectangular Occlusion Attack [231]
- AF: Adversarial Framing [253]
- SPA: Salt-and-Pepper Attack
- AT: Adversarial Training
- PGD-AT: Madry's AT formulation [150]
- BN: Batch Normalization [82]
- UDA: Unsupervised Domain Adaptation
- SFDA: Source-Free Domain Adaptation

Part I

Robust Computer Vision Against Adversarial Examples

Chapter 3

MultAV: Multiplicative Adversarial Videos

3.1 Motivation

The majority of existing adversarial attacks, including L_p-bounded attacks [60, 150, 201] and physical attacks [15, 191, 231] emphasize the *additive* attack approach where perturbation is *added* to input data; that is,

$$\mathbf{x}_{adv} = \mathbf{x} + \boldsymbol{\delta}_{adv}. \tag{3.1}$$

Lately, researchers have been studying some other attack types. Yang and Ji [248] presented a type of perturbation that multiplies input images by trained binary masks, but such perturbation is perceptible and requires complicated gradient estimators to optimize. Besides, their purpose is to regularize semi-supervised learning, so the adversarial attack strength is not evaluated. Some other recent works consider spatial attacks, which fool DNNs by small spatial perturbations [45, 234]; and coloring-based attacks, which generate adversarial examples by re-coloring the input images [113]. Solely delving into a handful of attack approaches would make the research community overlook many other possible adversarial examples that are threatening our machine learning systems. Hence, novel attacks need to be explored. On the other hand, adversarial examples for videos have not been investigated much in the literature, and all of these adversarial videos are additive attacks [93, 122, 228, 253] (see details in Chapter 2).

In this chapter, we propose a novel attack method, Multiplicative Adversarial Videos (MultAV), which can be applied to both ratio-bounded attacks and physically realizable attacks. Many coherent imaging systems such as Synthetic Aperture Radar (SAR) and ultrasound often suffer from multiplicative noises, known as speckle [62]. Inspired by this noise type, MultAV generates adversarial videos by *multiplying* crafted

noise with input examples:

$$\mathbf{x}_{adv} = \mathbf{x} \odot \delta_{adv}, \tag{3.2}$$

where $\odot$ denotes element-wise multiplication. MultAV can be imposed by different regularizations to keep the changes imperceptible using a new constraint on adversaries called Ratio Bound (RB). The ratio bound restricts the pixel-wise ratio of an adversarial example to an input example, corresponding to the L_p-norm of additive counterparts that restricts the pixel-wise difference. Furthermore, MultAV also applies to SPA (proposed in Chapter 2) and physically realizable attacks, where we consider the video version of ROA [231] and AF [253] in this chapter. We demonstrate that these attack types can be generated by the proposed multiplicative algorithm as well.

MultAV produces different perturbation distributions to the additive counterparts and thus challenges the defense approaches which are tailored to defending against additive adversarial attacks. Specifically, given MultAV examples of an attack type (the multiplicative version of this attack), the model adversarially trained against the additive counterpart is less robust than the model adversarially trained against MultAV directly. This gap also appears on Feature Denoising [239], a state-of-the-art defense, which demonstrates the threat of our MultAV and encourages more general and robust methods.

3.2 Proposed Method

We propose MultAV to fool video recognition systems. Recall that FGSM [60] builds the foundation for additive adversarial attacks, then PGD [150] extends FGSM to iterative versions for producing stronger attacks. Given a video data sample $\mathbf{x} \in \mathbb{R}^{F \times C \times H \times W}$ (F is the number of video frames, C is the number of channels, H

and W are height and width), ground-truth label $\mathbf{y}$, target model parameters θ and loss function $\mathcal{L}$, these iterative FGSM-based attacks generate adversarial examples $\mathbf{x}_{adv}$ by

$$\mathbf{x}^{t+1} = Proj^{L_\infty}_{\mathbf{x},\epsilon}\{\mathbf{x}^t + \alpha \cdot sign(\bigtriangledown_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t, \mathbf{y};\, \boldsymbol{\theta}))\}, \tag{3.3}$$

where α is step size, $t \in [0, t_{max} - 1]$ denotes the number of attacking iterations and thus $\mathbf{x} = \mathbf{x}^0$ and $\mathbf{x}_{adv} = \mathbf{x}^{t_{max}}$. $Proj^{L_\infty}_{\mathbf{x},\epsilon}\{\cdot\}$ projects its element into a L_∞ bound with perturbation size ϵ such that $|\mathbf{x}^{t+1} - \mathbf{x}| \leq \epsilon$. This L_∞-norm is the initial constraint used by the FGSM-based attacks. These attacks can also be bounded in L_2-norm:

$$\mathbf{x}^{t+1} = Proj^{L_2}_{\mathbf{x},\epsilon}\{\mathbf{x}^t + \alpha \cdot \frac{\bigtriangledown_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t, \mathbf{y};\, \boldsymbol{\theta})}{\| \bigtriangledown_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t, \mathbf{y};\, \boldsymbol{\theta})\|_2}\}, \tag{3.4}$$

where $Proj^{L_2}_{\mathbf{x},\epsilon}\{\cdot\}$ is a L_2-norm constraint with ϵ such that $\|\mathbf{x}^{t+1} - \mathbf{x}\|_2 \leq \epsilon$. In this case, the attacks take steps in the normalized gradient values instead of the sign of them.

MultAV belongs to gradient methods as well, and it can be formulated in a single step or iterative version. The iterative MultAV is defined as

$$\mathbf{x}^{t+1} = Proj^{RB-L_\infty}_{\mathbf{x},\epsilon_m}\{\mathbf{x}^t \odot \alpha_m^{sign(\bigtriangledown_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t,\mathbf{y};\boldsymbol{\theta}))}\}, \tag{3.5}$$

where α_m is the multiplicative step size, $Proj^{RB-L_\infty}_{\mathbf{x},\epsilon_m}\{\cdot\}$ performs projection with ratio bound ϵ_m such that $\max(\frac{\mathbf{x}^{t+1}}{\mathbf{x}}, \frac{\mathbf{x}}{\mathbf{x}^{t+1}}) \leq \epsilon_m$. MultAV takes the sign of the gradients as the exponent of α_m, so that $\mathbf{x}^t$ would be multiplied by either α_m or $1/\alpha_m$, which is corresponded to the additive counterparts added either α or $-\alpha$. The ratio bound is favorable to the multiplicative cases because the L_∞-norm would biased clip more perturbation in brighter pixels (having larger pixel values). Both the ratio bound and the L_∞-norm limit the perturbation maximum but in terms of addition and

multiplication, respectively. MultAV can also be extended to an L_2-norm variant:

$$\mathbf{x}^{t+1} = Proj_{\mathbf{x},\epsilon_m}^{RB-L_2}\{\mathbf{x}^t \odot \alpha_m^{\frac{\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t,\mathbf{y};\boldsymbol{\theta})}{\|\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t,\mathbf{y};\boldsymbol{\theta})\|_2}}\}, \tag{3.6}$$

where $Proj_{\mathbf{x},\epsilon_m}^{RB-L_2}\{\cdot\}$ a L_2-norm ratio constraint with ϵ_m such that $\|\frac{\mathbf{x}^{t+1}}{\mathbf{x}}\|_2 \leq (\epsilon_m + 1)$. Adding 1 is just an offset so that we can find a proper ϵ_m value easier. In this case, MultAV takes the normalized gradient values as the exponent of α_m.

SPA (Chapter 2) and the physically realizable ROA [231] and AF [253] examples can be generated by Eq. (3.3), but the perturbation is restricted in pre-defined regions, such as a rectangular, a framing and selected pixels. These attacks allow a large perturbation size since they can be perceptible. Similarly, their multiplicative versions can be produced by Eq. (3.5). MultAV is a general algorithm that applies to various attack types.

The perturbation generated by MultAV has distinct properties from that of additive adversarial examples. In particular, we can rewrite Eq. (3.5) as

$$\mathbf{x}^{t+1} = Proj_{\mathbf{x},\epsilon_m}^{RB-L_\infty}\{\mathbf{x}^t + \left[\mathbf{x}^t \odot (\alpha_m^{sign(\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t,\mathbf{y};\boldsymbol{\theta}))} - 1)\right]\}, \tag{3.7}$$

and rewrite Eq. (3.6) as

$$\mathbf{x}^{t+1} = Proj_{\mathbf{x},\epsilon_m}^{RB-L_2}\{\mathbf{x}^t + \left[\mathbf{x}^t \odot (\alpha_m^{\frac{\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t,\mathbf{y};\boldsymbol{\theta})}{\|\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t,\mathbf{y};\boldsymbol{\theta})\|_2}} - 1)\right]\}. \tag{3.8}$$

Eq. (3.7) and Eq. (3.8) indicate that the multiplicative perturbation can be treated as so-called *signal-dependent additive perturbation*, which involves the input data component in the additive perturbation.

Figure 3.1 shows different types of additive adversarial examples and MultAV

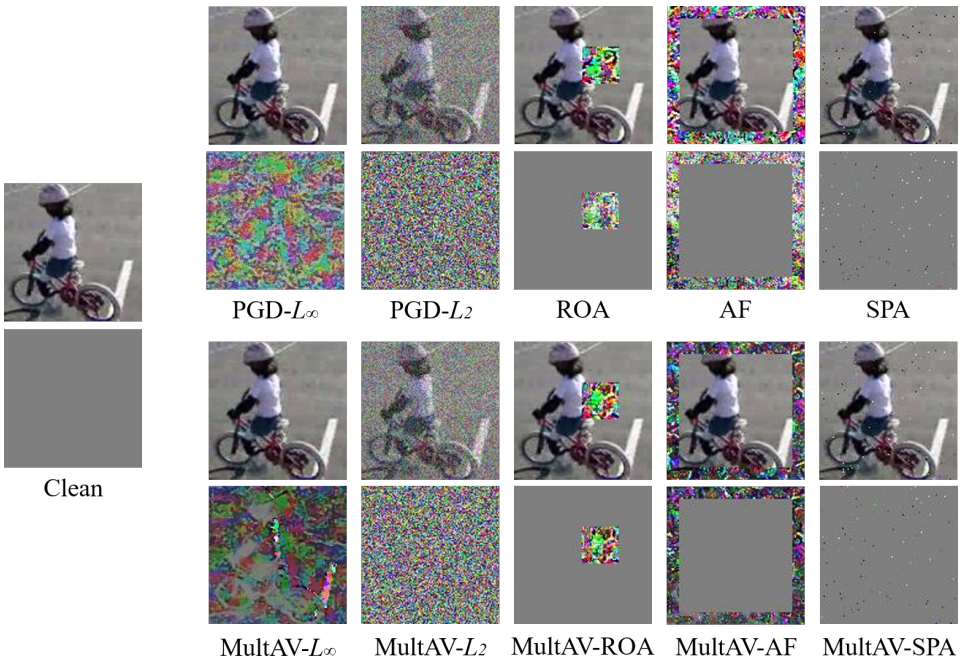

Figure 3.1: MultAV examples generated against 3D ResNet-18 (top); and difference maps ($15\times$ magnified for PGD-L_∞, PGD-L_2, MultAV-L_∞ and MultAV-L_2) between clean and each MultAV example (bottom). The first frame of the video is displayed here. Detailed attack settings are presented in Sec. 3.3.1.

examples. We can observe that the additive and the multiplicative perturbation distributions are different. Particularly, there is a clear object contour in the MultAV-L_∞ perturbation map, showing the signal-dependency of MultAV. The signal-dependent perturbation is more difficult to deal with since they are related to input data (signals). Because of the uniqueness of MultAV, the defenses tailored to resisting additive attacks may be ineffective, posing a new challenge to video recognition systems.

3.3 Experiments

We apply the proposed MultAV on L_∞-norm PGD [150], L_2-norm PGD, ROA [231], AF [253] and SPA (Chapter 2) attacks (MultAV-L_∞, MultAV-L_2, MultAV-ROA, MultAV-AF and MultAV-SPA), then evaluate them on AT-based state-of-the-art defense approaches. Furthermore, we look into the visualized feature maps under these adversarial attacks.

3.3.1 Exmerimental Setup

We conduct our experiments on the UCF101 dataset[198], an action recognition dataset consisting of 13,320 videos with 101 action classes. We use 3D ResNet-18 [65], a 3D convolution version of ResNet-18 [69], as our video classification network. All the models are trained or adversarially trained by SGD optimizer.

We adversarially train models against different MultAV types respectively and evaluate their robustness to these MultAV examples. The MultAV settings for both inference and AT in our experiments are described as follows:

- MultAV-L_∞: $\epsilon_m = 1.04$, $\alpha_m = 1.01$, and $t_{max} = 5$.
- MultAV-L_2: $\epsilon_m = 160$, $\alpha_m = 3.55$, and $t_{max} = 5$.
- MultAV-ROA: Rectangle size 30×30, $\epsilon_m = 1.7$, $\alpha_m = 3.55$, and $t_{max} = 5$.
- MultAV-AF: Framing width 10, $\epsilon_m = 3.55$, $\alpha_m = 1.7$, and $t_{max} = 5$.
- MultAV-SPA: 100 adversarial pixels on each video frame, $\epsilon_m = 3.55$, $\alpha_m = 1.7$, and $t_{max} = 5$.

We also adversarially train models against the additive counterparts and evaluate their robustness to MultAV. The additive counterparts for AT are set to similar attacking strength to their corresponding MultAV examples. The settings are as below:

- PGD-L_∞: $\epsilon = 4/255$, $\alpha = 1/255$, and $t_{max} = 5$.
- PGD-L_2: $\epsilon = 160$, $\alpha = 1.0$, and $t_{max} = 5$.
- ROA: Rectangle size 30×30, $\epsilon = 255/255$, $\alpha = 70/255$, and $t_{max} = 5$.
- AF: Framing width 10, $\epsilon = 255/255$, $\alpha = 70/255$, and $t_{max} = 5$.
- SPA: 100 adversarial pixels on each video frame, $\epsilon = 255/255$, $\alpha = 70/255$, and $t_{max} = 5$.

We test these attack approaches on 3D ResNet-18 with standard training and PGD-AT [150]. We also evaluate a feature denoising-based state-of-the-art defense approach [239]. Feature Denoising adds feature denoising blocks to an original network and performs the same AT protocol as PGD-AT on the entire network. We use the Gaussian version of non-local means denoising, which is their top-performing denoising operation. Feature Denoising is designed for only image data, so we extend it to video recognition tasks in two ways: 3D Denoise changes its operations to the 3D domain directly, and 2D Denoise performs the original 2D operations on each video example frame-by-frame. Following the deployment in [239], we insert 3D Denoise/2D Denoise after the conv2, conv3, conv4 and conv5 blocks of 3D ResNet-18.

Table 3.1: Video recognition accuracy (%) on the UCF101 dataset. The "Clean" column corresponds to the models that are trained and tested on clean data. The "Training" column refers to the data type for training, where "Mult" is a MultAV type corresponding to each of the last five columns, and "Add" is an additive counterpart of each of the last five columns.

Network	Clean	Training	MultAV-L_∞	MultAV-L_2	MultAV-ROA	MultAV-AF	MultAV-SPA
3D ResNet-18	76.90	Clean	7.19	2.67	2.30	0.26	4.02
3D ResNet-18	76.90	Mult	47.00	16.23	44.12	66.35	55.54
		Add	41.61	9.94	42.45	51.23	54.74
			(**-5.39**)	(**-6.29**)	(**-1.67**)	(**-15.12**)	(**-0.80**)
3D ResNet-18 + 3D Denoise	70.82	Mult	42.69	14.75	39.31	60.53	48.37
		Add	31.46	9.15	37.72	48.98	48.06
			(**-11.23**)	(**-5.60**)	(**-1.59**)	(**-11.55**)	(**-0.31**)
3D ResNet-18 + 2D Denoise	69.47	Mult	41.87	14.04	40.34	58.97	47.48
		Add	30.16	10.23	39.65	47.82	47.18
			(**-11.71**)	(**-3.81**)	(**-0.69**)	(**-11.15**)	(**-0.30**)

3.3.2 Evaluation Results

The experimental results are reported in Table 3.1. Doing AT makes models more robust to MultAV. However, we can see a serious robustness gap between the adversarially trained models against the additive counterparts (Add Model) and against MultAV directly (Mult Model). In particular, Add Model is less robust than Mult Model against MultAV, showing that the defenses tailored to defending against additive treat models fail to fully display their robustness under MultAV. Such a gap appears not only across all the considered MultAV types but also across the networks with and without Feature Denoising, which demonstrates the uniqueness and the threat of the proposed MultAV.

The gap size depends on MultAV types. It ranges from 5.39% to 11.71% on MultAV-L_∞, from 3.81% to 6.29% on MultAV-L_2, and from 0.69% to 1.67% on MultAV-ROA. The gap on MultAV-SPA is small. The reason is that the SPA perturbation is composed of scattered single pixels, and such noise distribution has no obvious difference between additive and multiplicative perturbation. Instead, the distribution

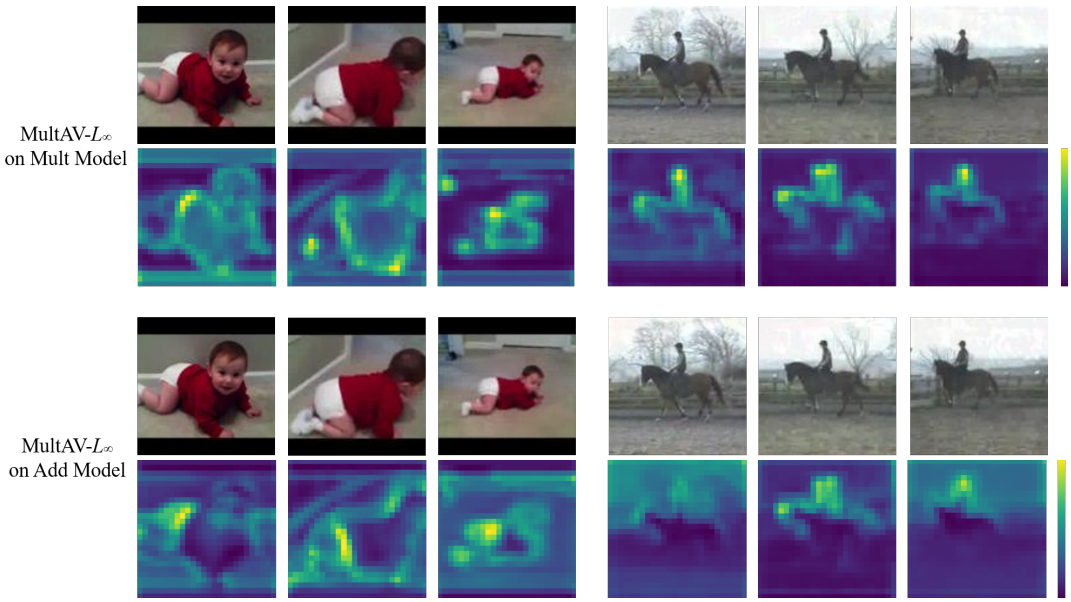

Figure 3.2: MultAV-L_∞ examples generated on Mult Model and Add Model (top); and their corresponding feature maps (bottom). Three frames of the video are displayed here.

differences are apparent in the other MultAV types. The most significant gap can be up to 15.12%, which appears on MultAV-AF.

On the other hand, Feature Denoising [239] does not perform well for the video recognition task. For both 3D Denoise and 2D Denoise, their clean data performance and adversarial robustness are degraded as compared with the original architecture. This indicates that an excellent defense for image recognition may be ineffective for videos. Our MultAV can be a good attack method motivating deeper exploration for adversarial robustness in the video domain.

3.3.3 Feature Map Visualization

Figure 3.2 shows the effect of adversarial perturbations on features. As can be seen from this figure, Mult Model is able to capture semantically informative content in

video frames. By contrast, Add Model's feature maps are blurred, which means Add Model is easier to be distracted by multiplicative adversarial perturbation and thus cannot focus on semantically informative regions. This visual result is consistent with the quantitive results in Table 3.1. The proposed MultAV poses a new and strong threat to video recognition models.

3.4 Summary

In this chapter, we propose a new attack method against video recognition networks, MultAV, which produces multiplicative adversarial videos having different noise distributions from the additive counterparts. It is a general multiplicative algorithm that applies to various attack types ranging from ratio-bounded attacks to physically realizable attacks. It challenges the defense approaches tailored to resisting additive adversarial attacks. Moreover, adversarial robustness in the video domain still lacks exploration. This clearly shows the threat of our MultAV. We hope this work will encourage the research community to look into more general and more powerful defense solutions for video recognition networks.

Chapter 4

Overcomplete Representations Against Adversarial Videos

4.1 Motivation

Several studies employ denoising-based methods to increase adversarial robustness. One stream uses denoising at the pre-processing stage to remove adversarial perturbations [91, 126, 177] (see details in Chapter 2). However, this stream is easily defeated by the adaptive attacks [5]. Other methods propose robust architectures, which contain denoising in network design instead of pre-processing and combine it with AT [150]. AT requires a larger network capacity than standard training [240], so designing network architectures having a high capacity to handle the difficult AT is important. FD [239] deploys a feature denoising block for their robust architecture design, which learns to remove perturbations at the feature level. Their method is only tested against a single attack type and only on the image data. On the other hand, defenses in the video domain are less explored. As discussed in Chapter 2, we are aware of few studies for detection or defense against video attacks (Chapter 6 and [90, 233]).

In this chapter, we follow the robust architecture stream and propose a novel Over-and-Under complete restoration network for Defending against adversarial videos (OUDefend). OUDefend learns the *overcomplete* representations [118] of input data against adversarial examples. Recall that FD [239] considers several classical denoising operations, including non-local means [17], bilateral filters [204], mean filters and median filters. Instead, we look into DNN-based algorithms for further improvements.

Traditionally, image/video restoration networks adopt an encoder-decoder architecture where the encoder first downsamples the input to a lower dimension spatially and the decoder then upsamples it back to the original dimension [250, 255]. Here

the receptive field of the filters in the deeper layers gets enlarged. Such a mechanism acquires *undercomplete* representations, which focus more on high-level features and global information but pay less attention to local details. In contrast, overcomplete representations are good at extracting meaningful low-level features and local information that are favorable for restoration [213]. Therefore, OUDefend consists of overcomplete and undercomplete braches to learn these two representation types respectively, and fuses their complementary features. We include OUDefend in target models as a feature restoration block and adversarially train the entire network end-to-end. The proposed method improves adversarial robustness against many different types of adversarial videos, including L_∞-norm PGD [150], L_2-norm PGD, MultAV (proposed in Chapter 3), ROA [231], AF [253] and SPA (proposed in Chapter 2). In particular, we show that FD is ineffective when applied to video data. Code is available at: https://github.com/shaoyuanlo/OUDefend

4.2 Proposed Method

We use video restoration to develop a robust architecture that has innate adversarial robustness for the problem of defense against adversarial videos. Previous DNNs used for restoration adopt a generic encoder-decoder architecture in which the encoder extracts an abstract version of input data while removing noise [250, 255]. To elaborate, they employ convolutional layers followed by max-pooling layers in the encoder and upsampling layers in the decoder. Such architecture is an example of undercomplete DNNs because the spatial dimension of the latent space representation is smaller than the inputs. As the receptive fields of filters increase after every max-pooling layer, the learned undercomplete representations collect more high-level features and global

context.

Overcomplete representations, in contrast to the undercomplete representations, were used as an alternative generic method for the representation of signals [118]. It involves using overcomplete dictionaries so that the number of basis functions is more than the number of input signal samples. This enables higher flexibility, leading to a robust representation of signals. Interestingly, DNNs employing overcomplete representations have not been explored much [213]. The overcomplete representations of visual data are able to acquire more meaningful low-level features and local context which are favorable for video restoration. As a result, we design an overcomplete network architecture to exploit the overcomplete representations where we project the input to a higher dimension spatially. In our overcomplete network, the receptive field gets constrained and so more low-level features and fine details are learned even in the deep layers compared to an undercomplete network. This happens as we use upsampling layers after each convolutional layer (instead of max-pooling in undercomplete networks) in the encoder which prevents the receptive field from enlarging in the deep layers. Furthermore, we fuse the over-and-under complete representations to fully gain their complementary advantages.

The architecture of the proposed OUDefend is illustrated in Figure 4.1. It has two branches: an overcomplete branch (O-branch) and an undercomplete branch (U-branch). O-branch has six $3\times3\times3$ convolutional layers in total, where the encoder and decoder both have three layers each. In the encoder of O-branch, each convolutional layer is followed by an upsampling layer, whereas in the decoder, each convolutional layer is followed by a downsampling layer. We employ interpolation for upsampling and max-pooling for downsampling. Skip connections [177] are used between the

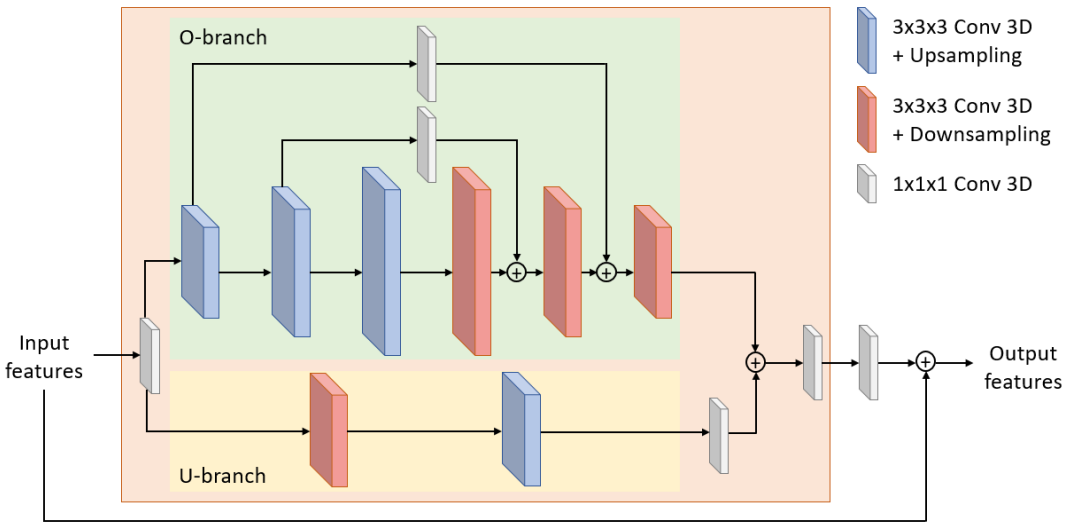

Figure 4.1: The proposed OUDefend architecture.

encoder and decoder for forwarding the features from early layers to the later layers thus helping in efficient gradient propagation. In addition, we propose using a $1\times 1\times 1$ convolutional layer in each skip connection. This layer learns and decides the most efficient earlier layer features that should be fused with the features of later layers. Although O-branch learns overcomplete representations that capture better low-level features than undercomplete representations, we note that undercomplete representations are also necessary as they leverage some high-level feature information which improves feature denoising. Thus, we propose having U-branch, a standard encoder-decoder structure [177, 7] with downsampling in the encoder and upsampling in the decoder. As U-branch is used as an auxiliary branch in OUDefend, we make it a lightweight 2-layer structure to reduce computational cost.

Next, we integrate the features from O-branch and U-branch. Before fusion, we have a $1\times 1\times 1$ convolutional layer at the end of U-branch to adjust the ratio of undercomplete representations before fusing them with their overcomplete counterparts.

We then increase the number of channels back by using another $1\times1\times1$ convolutional layer after the feature fusion of these two branches, and the outputs of this layer are the final restored feature maps. Finally, a $1\times1\times1$ convolutional layer and a residual connection are used. Since restoration may affect signals, it can maintain a balance between signal protection and noise suppression [239]. Furthermore, in order to keep the computational complexity low, at the beginning we pass the input features through a common $1\times1\times1$ convolutional layer to reduce the number of channels prior to feeding them to the two separate branches. All of the operations are 3D versions for processing video data.

4.3 Experiments

We evaluate the proposed OUDefend on six different types of adversarial videos: PGD-L_∞ [150], PGD-L_2, MultAV-L_∞ (Chapter 3), ROA [231], AF [253] and SPA (Chapter 2). These attack approaches range from additive attacks, multiplicative attacks to physically realizable attacks, so we can thoroughly test the adversarial robustness of OUDefend. We also present some analysis of the method by displaying the feature maps under attacks.

4.3.1 Experimental Setup

Our experiments are performed on the UCF-101 [198] dataset, an action recognition dataset composed of 13,320 videos with 101 action classes. 3D ResNet-18 [65], a 3D convolution version of ResNet-18 [69], is adopted as our backbone network. We attach an OUDefend architecture to 3D ResNet-18 as a feature restoration block after the conv4 block. All the networks are trained end-to-end using SGD optimizer. For

AT, we follow Madry's protocol [150].

For both inference and AT, the settings of the six considered attacks follow Chapter 6:

- PGD-L_∞: $\epsilon = 4/255$, $\alpha = 1/255$, and $t_{max} = 5$.
- PGD-L_2: $\epsilon = 160$, $\alpha = 1.0$, and $t_{max} = 5$.
- MultAV-L_∞: $\epsilon_m = 1.04$, $\alpha_m = 1.01$, and $t_{max} = 5$.
- ROA: Rectangle size 30×30, $\epsilon = 255/255$, $\alpha = 70/255$, and $t_{max} = 5$.
- AF: Framing width 10, $\epsilon = 255/255$, $\alpha = 70/255$, and $t_{max} = 5$.
- SPA: 100 adversarial pixels on each video frame, $\epsilon = 255/255$, $\alpha = 70/255$, and $t_{max} = 5$.

We compare our OUDefend with PGD-AT [150] and FD [239]. FD adds four feature denoising blocks to ResNet after the conv2, conv3, conv4 and conv5 blocks. We compare with their Gaussian version of non-local means denoising, which has two 1×1 convolutional layers for embeddings. It is their best-performing denoising operation. Since FD is designed for only image data, we extend it to the video domain in two ways: FD-A which replaces its operations by 3D versions directly, and FD-B which conducts the original 2D operations on videos frame-by-frame.

4.3.2 Evaluation Results

Table 4.1 reports our experimental results on the UCF-101 video recognition dataset. Both FD-A and FD-B fail to improve PGD-AT. In fact, these methods' performance on clean data drops as well. This indicates that their denoising structure does not

Table 4.1: Evaluation results (%) on UCF-101. Rows are defense methods. Columns are the number of model parameters, clean input data, and different attacks. No Defense is trained on clean data, and the others are trained on clean data or a specific attack (corresponding to the columns). Avg_{adv} is the average accuracy over the six attack types.

Method	Params	Clean	PGD-L_∞	PGD-L_2	MultAV	ROA	AF	SPA	Avg_{adv}
No Defense	33.0M	76.90	2.56	3.25	7.19	0.16	0.24	4.39	2.97
PGD-AT [150]	33.0M	76.90	33.94	35.05	47.00	41.29	74.81	55.99	48.01
FD-A [239]	33.7M	70.82	31.48	33.25	42.69	37.59	58.87	49.14	42.17
FD-B [239]	34.8M	69.47	30.19	32.65	41.87	38.22	58.74	49.14	41.80
OUDefend (Ours)	33.6M	**77.90**	**34.18**	**35.32**	**47.63**	**42.00**	**81.76**	**56.25**	**49.52**

work on video data and might degrade the quality of features. The reason may be that conventional image denoising operations cannot be generalized to video denoising very well, particularly when they are included as a part of a deep learning model, i.e., they are not compatible with video DNNs. Instead, OUDefend applies to 3D convolutional network architectures. It achieves the best adversarial robustness across all the six attack approaches we consider, showing its effectiveness ranges from additive attacks, multiplicative attacks to physically realizable attacks. Moreover, OUDefend's clean data performance is also better than the baseline architecture and FD, which demonstrates that adding OUDefend as a restoration block will not degrade the feature quality.

To show the importance of learning overcomplete representations in OUDefend, we build a variant that has only U-branch and thus learns undercomplete representations only. Under PGD-L_∞ attack, this variant obtains 33.15% accuracy, lower than OUDefend. This demonstrates the advantage of learning overcomplete representations.

On the other hand, OUDefend is a lightweight architecture that has only 0.6M parameters. It just accounts for 1.8% number of parameters when it is deployed in 3D

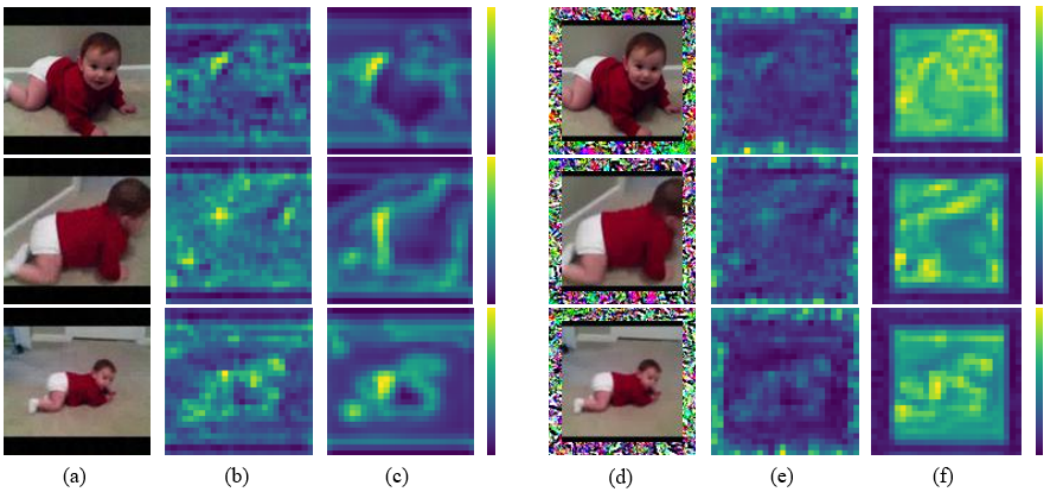

Figure 4.2: Feature maps after the conv2 block of No Defense and OUDefend under PGD-L_∞ and AF. No Defense is vanilla 3D ResNet-18 trained on clean data. OUDefend is adversarially trained, and here it is inserted after the conv2 block. Top to bottom: Three selected frames from a video. (a) PGD-L_∞ example. (b) No Defense's features under PGD-L_∞. (c) OUDefend's features under PGD-L_∞. (d) AF example. (e) No Defense's features under AF. (f) OUDefend's features under AF.

ResNet-18. It achieves the best performance and robustness with a fewer number of parameters than both FD-A and FD-B.

4.3.3 Feature Map Visualization

We visualize the feature maps of No Defense and OUDefend under the PGD-L_∞ and AF attacks in Figure 4.2. As can be seen, No Defense's features under PGD-L_∞ are noisy. Their activations are scattered over semantically trivial areas and thus fail to focus on informative content. The proposed OUDefend with AT leads to clearer features capturing fine details. Specifically, AT forces models to learn meaningful patterns in adversarial examples, and OUDefend further restores the perturbed features by leveraging over-and-under complete representations.

In the case of AF attacks, No Defense is misled to focus on the border of the video frames where the adversarial framings are located. Apparently, this area is semantically trivial. OUDefend with AT learns to ignore the border area and pay attention to the semantically meaningful regions. We can observe that the activations at the border area are almost zero. Therefore, the effectiveness of the proposed method is demonstrated by feature visualization as well.

4.4 Summary

We propose OUDefend, a new robust network architecture that exploits overcomplete representations to restore adversarial features. With an auxiliary undercomplete representation branch, it is able to balance local and global contexts by fusing these two representations. Adversarial robustness in the video domain is less explored, and our experiments show that the defenses tailored to images may be ineffective in videos. In contrast, OUDefend enhances robustness to many different types of adversarial videos, ranging from additive attacks, multiplicative attacks to physically realizable attacks.

Chapter 5

Error Diffusion Halftoning Against Adversarial Examples

5.1 Motivation

The existing empirical defense approaches can be categorized into two streams. The first one is based on AT [150]. This stream has been repeatedly validated as effective, especially under strong adaptive attacks in the challenging white-box setting [5, 18] (see details in Chapter 2). It has been widely used as a fundamental defense backbone [190, 231, 239]. The other stream deploys an image transformation at the pre-processing stage before inference to protect DNNs from adversarial effects [34, 42, 64, 91, 123, 244]. However, almost all attempts in this stream have been defeated under the white-box threat model. Raff et al. [173] stochastically combined a lot of image transformations to defend against white-box adaptive attacks, but failed to maintain the performance on clean images. The clean data performance is largely sacrificed owing to the multiple transforms.

In this chapter, we propose a novel defense method based on error diffusion halftoning [47, 87]. Different from most of the other image transformation-based defenses, it mounts resistance to adversarial examples even after accounting for the challenging adaptive attacks. To the best of our knowledge, this work is the first to leverage digital halftoning as the transformation for adversarial defense purposes.

Digital halftoning, which is sometimes referred to as spatial dithering, is a process of rendering a grayscale image into a binary image (i.e., black-and-white) [14, 43, 49, 211]. There are several commonly used halftoning algorithms. Thresholding quantizes each pixel value by comparing it with a fixed threshold. It is the simplest algorithm but results in poor rendering quality. Random dithering compares each pixel value with a random threshold to randomize quantization errors. This somewhat remedies the downside of thresholding. Ordered dithering creates a dither matrix to

Figure 5.1: (a) Original image. (b) Floyd-Steinberg halftone. One can clearly see the geometric structure present in the halftone.

turn the pixels black or white in a specified order, yielding better halftoning results. Error diffusion dithering [47, 87] contains an error diffusion mechanism that disperses quantization errors to neighboring pixels. This belongs to an area operation rather than a simple pointwise operation and can mitigate visual artifacts. We choose error diffusion dithering as our halftoning transformation and deploy it at the pre-processing stage for defense. Specifically, Floyd-Steinberg dithering [47] is used in our approach. Figure 5.1 shows a sample image and the corresponding Floyd-Steinberg halftone. As can be seen from this figure, even though the halftone only consists of black and white dots, it maintains the overall structure of the object present in the image.

Error diffusion halftoning quantizes pixel values to filter out adversarial perturbation, and the error diffusion mechanism can weaken adaptive attacks. Moreover, spreading quantization errors produces higher halftoning quality and thus maintains better accuracy on clean data. The proposed method significantly improves robustness against different adversarial attacks, including PGD [150] and MultAV (proposed in

Chapter 3), under the white-box setting. In the meantime, it is able to achieve good clean data performance. Although most of the image transformation-based defenses have been proven ineffective, we show that this stream is still worth exploring. Code is available at: https://github.com/shaoyuanlo/Halftoning-Defense

5.2 Proposed Method

We use error diffusion halftoning [47, 87] as the image transformation for the adversarial defense purpose. The key idea is to quantize each pixel in the raster order (from left to right, top to bottom) one-by-one, and spread the quantization error to the neighboring pixels. Beginning with the top-left pixel, the pixel value is binarized by thresholding, then the quantization error is dispersed to neighboring pixels using predefined weights. Following the raster-scan indexing scheme, the procedure continues until the bottom-right pixel has been transformed. More precisely, let us consider an input image I with pixel values $\in [0,1]$, and an error filter h. For each pixel $I(i,j)$ in I with the raster order, it pulls the error forward as:

$$\hat{I}(i,j) = I(i,j) + \sum_{m,n \in S} h(m,n)e(i-m,j-n). \tag{5.1}$$

Next, $\hat{I}(i,j)$ is quantized to a binary value:

$$Q(i,j) = u(\hat{I}(i,j) - \theta), \tag{5.2}$$

where $u()$ is a unit step function with a threshold $\theta = 0.5$. The pixel's quantization error is calculated as:

$$e(i,j) = \hat{I}(i,j) - Q(i,j). \tag{5.3}$$

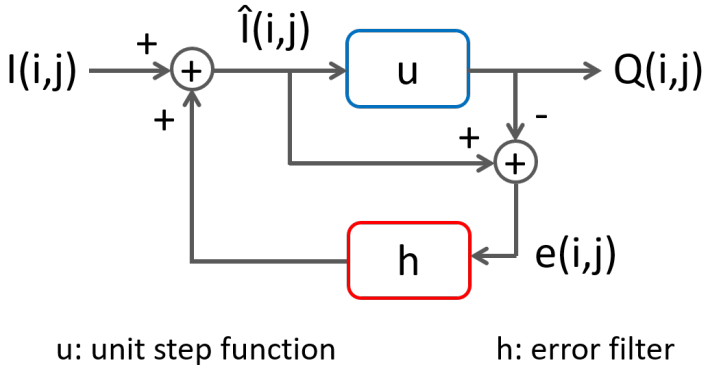

Figure 5.2: Error diffusion halftoning scheme.

Then, this error is pushed ahead, and the next pixel in the raster order pulls the errors, repeating from Eq. (5.1) to Eq. (5.3) until the last pixel. Figure 5.2 summarizes this procedure [14].

In this chapter, we implement error diffusion halftoning by Floyd-Steinberg dithering [47] because of its efficiency and fine-grained results. The Floyd-Steinberg error filter is defined as:

$$h_{FS} = \frac{1}{16}\begin{bmatrix} 0 & * & 7 \\ 3 & 5 & 1 \end{bmatrix}, \tag{5.4}$$

where $*$ denotes the pixel being scanned currently, and it only disperses errors to adjacent pixels. The weights are zeros for the pixels that have been scanned, so the error diffusion does not go backward with respect to the raster order. Alg. 1 describes this algorithm in detail. For color images, these operations are performed for each channel independently.

We deploy Floyd-Steinberg dithering as an image transformation at the preprocessing stage. To elaborate, let x_{adv} be an adversarial example and T_{FS} be the

Algorithm 1: Floyd-Steinberg dithering

Result: Output halftone Q
Given an input image I with pixel values $\in [0, 1]$,
for *i from top to bottom* **do**
  for *j from left to right* **do**
    $oldValue = I[i][j]$
    if $oldValue > 0.5$ **then**
      $newValue = 1$
    else
      $newValue = 0$
    end
    $Q[i][j] = newValue$
    $error = oldValue - newValue$
    $I[i+1][j] \mathrel{+}= error \times 7/16$
    $I[i-1][j+1] \mathrel{+}= error \times 3/16$
    $I[i][j+1] \mathrel{+}= error \times 5/16$
    $I[i+1][j+1] \mathrel{+}= error \times 1/16$
  end
end

Floyd-Steinberg dithering, then the input of the target model is $T_{FS}(x_{adv})$. Floyd-Steinberg dithering can invalidate the adversarial variations of pixel values and destroy the structure of adversarial perturbations through the quantization operation. Moreover, the error diffusion mechanism repeatedly updates the values of the neighboring pixels in the raster order. This makes the adaptive attacks hard to identify the mapping between the original image and the corresponding halftone, so BPDA [5] would be difficult to approximate the gradients accurately to generate strong adversarial examples. Therefore, Floyd-Steinberg dithering allows us to mitigate adversarial effects in advance, then feed an adversary-free image to the target model for protection. We employ PGD-AT [150] to train the model end-to-end. That is, the halftoning transformation is included in the training process so that the model can learn to recognize halftones with adversarial patterns. On the other hand, spreading quantization errors

produces better halftoning quality and tends to enhance edges and object boundaries in an image, which are favorable to image recognition tasks. Furthermore, compared to other complicated error diffusion halftoning algorithms, Floyd-Steinberg dithering only diffuses errors to the adjacent pixels, so it saves computational costs. In short, the proposed method takes adversarial robustness, clean data performance, and efficiency into consideration, achieving an excellent balance between these three indicators.

5.3 Experiments

We evaluate our method on four attack types in the white-box setting: PGD-L_∞ [150], PGD-L_2, Mult-L_∞ and Mult-L_2 (Chapter 3). Mult attack is originally designed for adversarial videos (MultAV), and we apply it to generate adversarial images. These attacks include both additive and multiplicative attacks. Clean images are also tested. We compare the performance of our approach with four image transformation-based defenses: Gaussian blur, non-local means [17], JPEG compression [34, 42] and bit-depth reduction [64, 244]. Finally, a deep analysis is provided.

5.3.1 Experimental Setup

We conduct experiments on CIFAR-10 [103], an image classification dataset that consists of 60,000 images with size 32×32 from 10 classes. We adopt ResNet-18 [69] as the backbone network. All the models are trained by the SGD optimizer. We follow Madry's protocol [150] for AT.

The settings of the four considered attacks for both inference and AT follow Chapter 3. PGD-L_∞: $\epsilon = 8/255$, $\alpha = 3/255$. PGD-L_2: $\epsilon = 1.0$, $\alpha = 3.0$. Mult-L_∞: $\epsilon_m = 1.08$, $\alpha_m = 1.03$. Mult-L_2: $\epsilon_m = 1.3$, $\alpha_m = 1.03$. $t_{max} = 5$ for all the attacks.

Table 5.1: Evaluation results (%) on CIFAR-10. Rows are defense methods, and columns are input types. On the standard training track, models are trained on clean data. On the AT track, the "Clean" column is that models are trained on PGD-L_∞ data but tested on clean data. For the other columns, models are trained on a specific attack type (corresponding to the columns). Avg_{adv} and Avg_{all} denote average accuracies over the four attacks and over all the five data types, respectively.

Method	Training	Clean	PGD-L_∞	PGD-L_2	Mult-L_∞	Mult-L_2	Avg_{adv}	Avg_{all}
Vanilla	Standard training	**94.03**	0.01	0.20	0.05	0.01	0.07	18.86
Gaussian blur		90.17	0.20	1.34	0.17	0.05	0.44	18.39
Non-local means		88.66	0.02	0.49	0.03	0.00	0.14	17.84
JPEG compression		90.06	2.97	4.82	1.81	0.22	2.46	19.98
Bit-depth reduction		78.87	**15.26**	10.84	**10.79**	**4.52**	**10.35**	**24.06**
Halftoning (Ours)		88.57	9.53	**11.98**	5.54	1.07	7.03	23.34
Vanilla	Adversarial training	83.31	51.15	50.68	54.10	40.29	49.06	55.91
Gaussian blur		75.96	44.59	47.12	45.07	32.48	42.32	49.04
Non-local means		75.47	44.67	45.29	16.59	14.53	30.27	39.31
JPEG compression		24.97	38.99	43.72	59.15	44.72	46.65	42.31
Bit-depth reduction		71.66	47.34	42.40	48.50	41.63	44.97	50.31
Halftoning (Ours)		**84.37**	**60.01**	**56.56**	**67.37**	**88.44**	**68.10**	**71.35**

For the four compared defenses, we set the hyper-parameters as follows: The kernel of Gaussian blur is 5×5 with $\sigma = 1.5$; non-local means is with the Gaussian version; the JPEG compression level is $30/100$; and the bit-depth reduction quantizes pixel values to 1-bit for each channel. Because JPEG compression, bit-depth reduction and halftoning cause obfuscated gradients, we employ BPDA [5] to mount adaptive attacks for evaluating these defenses. The identity function is used as a surrogate function to approximate the gradients.

5.3.2 Evaluation Results

Table 5.1 reports our experimental results. In the case of standard training, both Gaussian blur and non-local means provide no resistance to any type of adversarial attacks under the white-box threat model. JPEG compression shows somewhat effectiveness. Only bit-depth reduction and halftoning obviously improve the robustness to

all of the considered attacks. On the other hand, image transformation-based defenses usually suffer from a drop in clean data accuracy since the transformations degrade the semantic information. Particularly, bit-depth reduction obtains much lower clean data performance. Instead, halftoning is able to enhance the robustness and preserve the performance simultaneously.

AT has become the backbone of advanced defense approaches. We combine these image transformation-based defenses with AT to pursue better robustness. As can be seen from Table 5.1, AT makes great improvements. However, Gaussian blur, non-local means and bit-depth reduction fail to improve upon the vanilla AT baseline but decrease the performance. Learning features with adversarial patterns is more difficult, so AT requires higher model capability [239]. These three transformations degrade the model capability and thus cannot handle AT, resulting in worse robustness. Furthermore, Gaussian blur and non-local means are especially vulnerable to Mult attacks, indicating these two image denoising transformations are unable to deal with the multiplicative adversarial perturbations. JPEG compression is useful for Mult attacks but ineffective in PGD attacks. In contrast, the proposed method significantly improves the robustness over all the four considered attacks.

Similarly, AT decreases clean data accuracy [236], and the image transformation-based defenses make further degradation. In particular, JPEG compression obtains very low clean data performance when it is trained adversarially. Instead, halftoning's performance drops slightly and achieves the highest clean data accuracy in the AT case, which is even better than the vanilla model. This shows that the halftoning defense is able to handle the difficult AT and can generalize to clean images. These results demonstrate the proposed method is a preferred defense method that can improve

robustness against different types of attacks under the white-box setting. Concurrently, it maintains good clean data performance.

In addition, we submit this work to the DARPA GARD [40] project with a third-party evaluator. Their results show that on the UCF-101 [198] dataset, the proposed Halftoning with combining AT achieves 92% accuracy under the attacks of PGD with $\epsilon = 4/255$ and Masked PGD with 20% occlusion. This is the best result for action recognition over all the submissions of this project.

5.3.3 Analysis

We display some of PGD-L_∞ adversarial examples and their corresponding transformed images in Figure 5.3. All the transformations lose information to a certain extent, so their clean data performance drops. Particularly, bit-depth reduction produces very coarse images. Instead, halftoning can filter out adversarial perturbations but still maintains highly recognizable image quality.

Figure 5.3 also shows the visualized differences between the features of clean images and the features of the corresponding transformed PGD-L_∞ examples. The quantitative values of such differences are compared in Figure 5.4. The vanilla model has the largest difference, which means the features are largely changed when the image is adversarially perturbed and thus causes a wrong prediction. Gaussian blur, non-local means and JPEG compression repress the differences yet insufficiently, so their robustness is still poor. Bit-depth reduction obtains the smallest difference, but its transformed images are too coarse to recognize accurately. In contrast, halftoning also attains a small difference, showing its features are not easily affected by adversarial perturbation. In the meantime, its image quality is highly recognizable. Hence, our

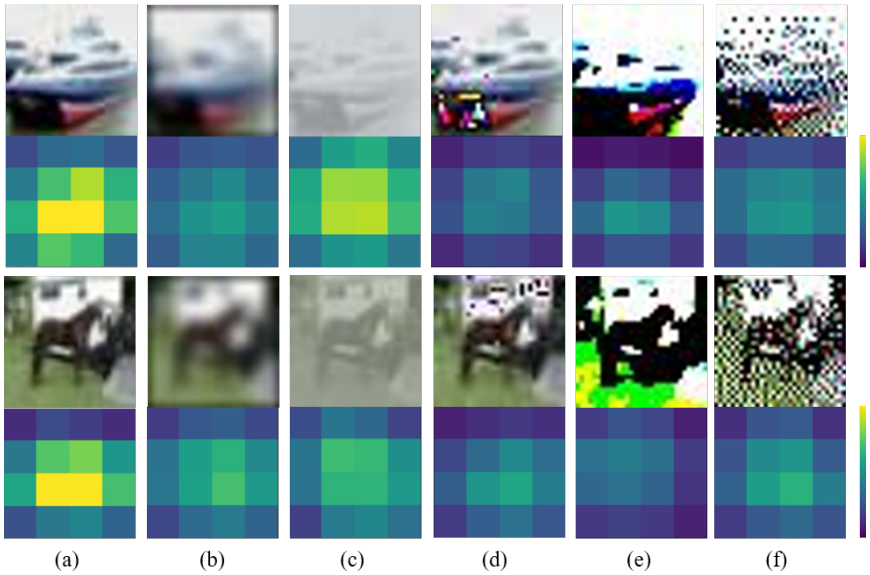

Figure 5.3: Transformed PGD-L_∞ adversarial examples, and the visualized differences between the features of clean images and the features of corresponding transformed PGD-L_∞ examples. The features (size 8×8) are from the last conv layer of ResNet-18. (a) Vanilla. (b) Gaussian blur. (c) Non-local means. (d) JPEG compression. (e) Bit-depth reduction. (f) Halftoning. Models are with standard training.

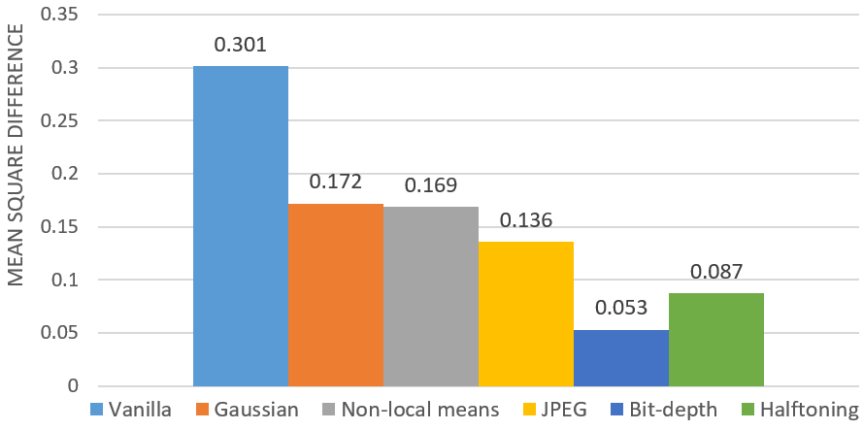

Figure 5.4: Mean square differences between the features of clean images and the features of corresponding transformed PGD-L_∞ examples. The features are from the last conv layer of ResNet-18. Models are with standard training. The values are the averages of the CIFAR-10 test set.

method can achieve good robustness and performance simultaneously.

5.4 Summary

In this chapter, we propose a novel image transformation-based defense method by using Floyd-Steinberg halftoning. The 1-bit quantization and error diffusion mechanisms can remove adversarial perturbations and weaken adaptive attacks. Furthermore, the proposed method's ability to produce high-quality halftones guarantees good clean data performance. Although the majority of the image transformation-based defenses have been shown to be ineffective under the white-box threat model, our method is able to greatly improve adversarial robustness. We show that this defense stream is still promising and worthy to explore.

Chapter 6

Defending Against Multiple and Unforeseen Adversarial Videos

6.1 Motivation

Existing AT approaches usually lead to performance degradation on clean data [209, 256]. Xie et al. [236] indicated that this problem is due to the distribution mismatch between clean and adversarial examples. In order to deal with this issue, they leveraged an auxiliary BN layer [82] to disentangle the two distributions. In addition, most existing AT techniques are tailored to one specific perturbation type, e.g., a certain L_p-norm perturbation [150, 174, 229] or physically realizable attacks [231]. A model trained on a specific attack can improve its robustness to that particular attack but often fails to defend when presented with a sample that is perturbed by a different type of attack [192]. Although there have been several attempts aim to resist multiple attack types, they usually do not consider physically realizable attacks, unforeseen attacks, or how well they perform on clean images [114, 127, 153, 205]. In a real-world application, the input data could be clean (i.e., unattacked), adversarial, or even attacked with a novel attack that the network has never seen before.

On the other hand, most recent research in this area has focused on static images. Generating adversarial examples and defense methods for videos is relatively less explored. As discussed in Chapter 2, although a few recent works have extended adversarial attacks to videos [93, 122, 227, 228, 253], we are aware of few studies so far that delve into detecting or defending against adversarial videos (Chapter 4 and [90, 233]).

In this chapter, we propose MultiBN, which is one of the first defense methods for defending against adversarial videos and considering the accuracy on clean samples as well as the robustness to multiple and unforeseen perturbations. Specifically, we consider four of the most significant types of attacks: PGD [150], ROA [231], AF

Figure 6.1: Illustration of the four types of adversarial videos we consider. Three video frames from the UCF-101 [198] dataset are displayed here.

[253] and SPA (proposed in Chapter 2). Figure 6.1 gives an illustration of these attacks on video frames. PGD and ROA are originally designed to attack images. We extend these to videos by perturbing each frame and unveil that video recognition models are also vulnerable to these attacks. SPA is a new video attack we design in Chapter 2, which looks like salt-and-pepper noise. PGD and SPA belong to the L_p-bounded attack group, while ROA and AF belong to the physically realizable attack group. We select one from each group as the known attack type (PGD and ROA) and leave the others as the unforeseen attack type (AF and SPA), where only the known attacks are used for AT. MultiBN aims to defend against all of these attack types while retaining the performance on clean samples simultaneously.

We first demonstrate that training a model on a specific attack type can gain robustness to that attack and somewhat to another attack in the same group, but typically

cannot defend against the attacks in another group. Training models on multiple attack types together (*multi-perturbation training*) improves *multi-perturbation robustness*, yet the accuracy on clean samples is sacrificed. This is mainly due to the distribution mismatch among clean and different types of adversarial examples. We assume that the attacks in the same group have a relatively similar distribution. Therefore, inspired by [240, 236], the proposed MultiBN employs multiple BN branches in a single network: for the clean, L_p-norm and physically realizable attack examples, individually. Each BN branch is responsible for learning the distribution of a specific type of examples, which can offer more accurate distribution estimations for these types. Because BN is a lightweight component included in common DNNs, using multiple BN branches causes only minor parameter increases and computational overhead. MultiBN also contains a BN selection module, which detects the attack type of an input video and sends it to the corresponding BN branch, so the entire MultiBN is fully automatic and allows end-to-end training. Compared to existing AT and multi-perturbation training approaches, MultiBN achieves stronger adversarial robustness against multiple, more diverse, and even unforeseen perturbations, while retaining higher accuracy on clean samples. Moreover, MultiBN demonstrates effectiveness in the image domain as well. An extensive analysis showing the properties of the multiple BN structure is also presented. As one of the first studies of multi-perturbation robustness for videos, this chapter provides baseline results that broadly cover multiple attack types, datasets and target models, for this problem. We hope that these baselines will be useful to other researchers and the adversarial robustness community. Code is available at: https://github.com/shaoyuanlo/MultiBN

The main contributions of this chapter are summarized as follows:

- We propose a novel adversarial defense method, MultiBN, based on a multiple BN structure and a BN selection module. To the best of our knowledge, this is the first defense against multiple and unforeseen adversarial videos.
- The proposed MultiBN achieves both stronger multi-perturbation robustness and better clean sample performance than existing multi-perturbation training approaches. This holds true on different datasets and target models.
- We provide extensive analysis to study the properties of the multiple BN structure under various conditions.
- We provide comprehensive baseline results for multi-perturbation robustness in the video domain. These baselines broadly cover multiple attack types, threat models, datasets, and target networks.

6.1.1 Related Work

Xie et al. [240, 236] demonstrated that proper normalization management is important for enhancing robustness and even performance. Although our work is inspired by [236], [236] aims to leverage the AT technique to improve image recognition performance on clean data. It does not consider the model's adversarial robustness, and its model is not applicable to the multi-perturbation robustness problem.

Several studies focus on multi-perturbation robustness. Tramèr et al. [205] investigated adversarial robustness to multiple perturbations, including L_p-bounded attacks and rotation-translation attacks. They provided AVG and MAX AT schemes. Maini et al. [153] incorporated multi-perturbation models into a single attack by Multi Steepest Descent (MSD). MSD is robust to different L_p-bounded attacks. Nevertheless,

these studies do not take potential unforeseen attack types and clean images into consideration. Laidlaw et al. [114] adversarially trained a target model by Neural Perceptual Threat Model (NPTM), showing good resistance to L_p-bounded attacks and spatial attacks. However, its robustness cannot generalize to physically realizable attacks. Lin et al. [127] aimed to defend against L_p and non-L_p attacks, but they require a pre-constructed On-Manifold dataset, which is too expensive for practical uses. Our MultiBN manages normalization with low costs to enhance the robustness to multiple, more diverse, and even unforeseen perturbations, while retaining higher accuracy on clean images simultaneously.

6.2 Preliminary

6.2.1 Multiple and Unforeseen Adversarial Videos.

In this chapter, we construct four types of video attacks: L_∞-norm PGD [150], ROA [231], AF [253], and the new SPA attack (see Figure 6.1). Among them, L_∞-norm PGD and SPA (L_0-norm) belong to the L_p-bounded attacks; ROA and AF belong to the physically realizable attacks. In our experiments, we set PGD and ROA as the known attack types available for AT, while AF and SPA are used as unforeseen attack types used only during inference. We aim to defend against multiple adversarial video types, including L_p-bounded and physically realizable attacks as well as known and unforeseen attacks. All of these attacks are set to untargeted since the untargeted attack is considered more difficult to resist than the targeted attack.

6.2.2 Adversarial Training and Multi-perturbation Training.

The proposed MultiBN is based on AT and multi-perturbation training. We briefly review AT and state-of-the-art multi-perturbation training schemes to describe the preliminary formulation of our method.

To begin with, we recall the objective function for training a DNN model:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\mathcal{L}(x, y; \theta)\right], \tag{6.1}$$

where x is a clean training sample with ground-truth label y in the training set $\mathbb{D}$, θ is model parameters, and $\mathcal{L}$ denotes the training loss. PGD-AT [150] applies the min-max optimization and trains models exclusively on adversarial examples:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\max_{\delta\in\mathbb{S}} \mathcal{L}(x+\delta, y; \theta)\right], \tag{6.2}$$

where δ denotes an adversarial perturbation that is within a bounded perturbation set $\mathbb{S}$. AdaProp [236] aims to improve the performance on clean samples and trains the model with a mixture of clean data and adversarial examples as follows: [60, 110]:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\mathcal{L}(x, y; \theta) + \max_{\delta\in\mathbb{S}} \mathcal{L}(x+\delta, y; \theta)\right]. \tag{6.3}$$

Note that AdaProp is not designed for multi-perturbation robustness. TRADES [256] uses an alternative objective function for AT:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\mathcal{L}(x, y; \theta) + \max_{\delta\in\mathbb{S}} \mathcal{L}(x+\delta, f(x); \theta)\right], \tag{6.4}$$

where $f(x)$ is the output vector of the target model with a Softmax operator. In other words, TRADES replaces y with $f(x)$ to compute the cross-entropy loss of adversarial examples.

Regarding multi-perturbation robustness, Tramèr et al. [205] introduced two AT strategies: AVG strategy and MAX strategy. AVG trains on all types of adversarial examples simultaneously and optimizes these adversarial losses together as follows:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\sum_{i=1}^{N} \max_{\delta_i \in \mathbb{S}_i} \mathcal{L}(x+\delta_i, y; \theta) \right], \tag{6.5}$$

where N is the number of perturbation types. MAX considers the worst-case attack. It trains on the strongest adversarial example that obtains the maximum loss among all types of attacks:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\mathcal{L}(x+\delta_k, y; \theta) \right], \tag{6.6}$$

where

$$\delta_k = \arg\max_{i\in[1,N]} \left[\max_{\delta_i \in \mathbb{S}_i} \mathcal{L}(x+\delta_i, y; \theta) \right], \tag{6.7}$$

which denotes the strongest type of attack. MSD [153] maximizing the worst-case loss over all the considered perturbations at each projected steepest descent step to construct a single perturbation. This can be described as the follows:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\mathcal{L}(x+\delta_{MSD}, y; \theta) \right], \tag{6.8}$$

where δ_{MSD} is the constructed single perturbation.

6.3 Proposed Method

In real-world applications, the input data could be clean, adversarial, or even attacked with a novel attack that the network has never seen before. Hence, it is important to design a defense solution that can resist multiple known and unforeseen perturbations

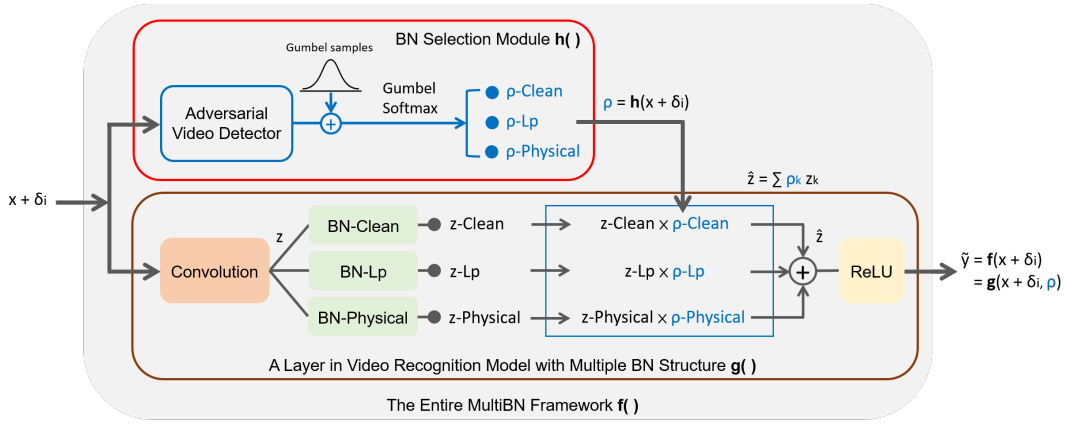

Figure 6.2: Overview of MultiBN, the proposed adversarial defense framework. Every batch normalization layer of the video recognition model is replaced by a multiple BN structure, where this figure illustrates only one layer for simplicity. $x + \delta_i$: an input of a specific type adversarial example, z_k: the k-th BN branch's output features, ρ_k: a ratio factor to weight the k-th BN branch's output features, $\tilde{y}$: prediction.

while retaining the performance on clean samples. The proposed method, MultiBN, is based on a multiple BN structure and a BN selection module. Figure 6.2 gives an overview of MultiBN.

6.3.1 Multiple Batch Normalization Structure

AT on a single perturbation type is generally weakly robust to the other types of attacks. On the other hand, most state-of-the-art DNNs contain BNs [82] in their architecture to normalize input features, which improves performance [65, 69]. However, owing to the different distributions among multiple perturbation types, BNs suffer from the distribution mismatch when multi-perturbation training is conducted, and thus fail to gain promising multi-perturbation robustness. To address this problem, we deploy multiple BN branches into each BN layer of the target model and keep the rest of the parts unchanged, i.e., still a single network [236, 240]. Clean data and each perturbation type used for training are assigned an individual BN branch. Since BN is

a lightweight component, multiple BN branches cause only minor parameter increases and computational overhead.

Ideally, each BN branch is responsible for estimating the assigned single or a family distribution(s), and thus can properly disentangle multiple distributions. Consider AT as a min-max optimization problem [150], for forward and backward passes, we can manually make each perturbation type attack the target model through its assigned BN branch at the inner maximization step. For the outer minimization step, we send clean inputs or the generated adversarial examples to their corresponding BN branch as well. The ideal objective function can be defined as follows:

$$\theta^* = \arg\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathbb{D}} \left[\mathcal{L}(x, y; \theta^c, \theta_0^b) + \sum_{i=1}^{N} \max_{\delta_i \in \mathbb{S}_i} \mathcal{L}(x + \delta_i, y; \theta^c, \theta_i^b) \right], \quad (6.9)$$

where θ^c is convolution parameters, θ_i^b is the BN parameters of the i-th data type, and $\theta = \theta^c + \sum_{i=0}^{N} \theta_i^b$ denotes all the model parameters.

In practical scenarios, DNNs should provide robustness against unforeseen attacks. An exhaustive investigation is too expensive; instead, we can summarize different attack types into several groups based on their distributions, then build a BN branch for each group. In our case, we deploy three BN branches for clean data, L_p-bounded attacks and physically realizable attacks, respectively (see Figure 6.2). Typically, AT on a strong attack has better robustness [150], where PGD and ROA are good representatives of L_p-bounded attacks and physically realizable attacks, respectively. Therefore, we train a target model on clean, PGD, and ROA examples using the 3-BN structure with Eq. (6.9), where $N = 2$.

6.3.2 Batch Normalization Selection Module

At inference time, we cannot control the data flow, so the input data have to pass through the corresponding BN branch automatically. To this end, we propose a BN selection module based on an adversarial video detector and a Gumbel-Softmax operator [86, 149] (see Figure 6.2). The adversarial video detector is achieved by a $(N+1)$-class video classification model, where N is the number of attack types used for training. It is trained to not only identify whether an input video is clean or adversarial but also recognize the attack types. In our case, the detector is trained with $N = 2$ on clean, PGD and ROA examples to recognize clean data, L_p-bounded attacks and physically realizable attacks. Hence, unforeseen perturbations would also be classified into the most similar attack group.

Intuitively, we can build a switch module to send the input to the proper BN branch according to its detection result. However, the *argmax* operation, which applies to the adversarial video detector's logits for BN branch selection, is not differentiable. This makes end-to-end training infeasible. To address this issue, we leverage the Gumbel-Softmax trick to allow the gradients to backpropagate through a discrete sampling process [243]. Specifically, we approximate *argmax* by the differentiable and continuous Gumbel-Softmax function, which is defined as follows:

$$\rho_i = \frac{exp((\log\ \pi_i + G_i)\ /\ \tau)}{\sum_{j=1}^{K} exp((\log\ \pi_j + G_j)\ /\ \tau)}, \tag{6.10}$$

where $\pi_1, ..., \pi_K$ are the adversarial video detector's logits, $G_1, ..., G_K$ are i.i.d. Gumbel samples, τ is the Softmax temperature, and $K = 3$ in our case. Next, $\rho_1, ..., \rho_K$ are

used as ratio factors to weight each BN branch's output features:

$$\hat{z} = \sum_{i=1}^{K} \rho_k z_k, \tag{6.11}$$

where $z_1, ..., z_K$ are each BN branch's output features, and $\hat{z}$ is the weighted feature that would be the input of the next network component (see Figure 6.2). In this way, the correct BN branch's output feature z_* would dominate $\hat{z}$, making $\hat{z}$ a good approximation of z_*.

6.3.3 The Entire Framework

With the BN selection module, MultiBN can operate automatically during inference without manual control, and it enables end-to-end training since the entire framework is differentiable. Let $f(\,)$ be the entire MultiBN framework, $g(\,)$ be the video recognition model with the multiple BN structure, and $h(\,)$ be the BN selection module (see Figure 6.2). Consider an input of a specific type adversarial example $x + \delta_i$, the entire end-to-end pipeline can be described as follows:

$$\tilde{y} = f(x + \delta_i;\, \theta^c, \theta^b, \theta^{det}) = g(x + \delta_i, h(x + \delta_i;\, \theta^{det});\, \theta^c, \theta^b), \tag{6.12}$$

where $\tilde{y}$ is the prediction, θ^c is $g(\,)$'s convolution parameters, $\theta^b = \sum_{i=0}^{N} \theta_i^b$ ($N = 2$ here, see Sec. 6.3.1) is $g(\,)$'s BN parameters in all the BN branches, and θ^{det} denotes the parameters of the adversarial video detector in the BN selection module $h(\,)$. $h(\,)$ outputs $\boldsymbol{\rho} = [\rho_1, ..., \rho_K]$ defined in Sec. 6.3.2, i.e., $\boldsymbol{\rho} = h(x + \delta_i;\, \theta^{det})$.

Then, the end-to-end training objective can be written as follows:

$$\theta^* = \arg\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathbb{D}} \Big[\mathcal{L}(x, y; \theta) + \lambda \cdot \mathcal{L}(x, y^{det}; \theta^{det}) + \sum_{i=1}^{N} \big(\max_{\delta_i \in \mathrm{S}_i} \mathcal{L}(x + \delta_i, y; \theta) + \lambda \cdot \mathcal{L}(x + \delta_i, y^{det}; \theta^{det}) \big) \Big], \tag{6.13}$$

where $\theta = \theta^c + \theta^b + \theta^{det}$ contains all the entire MultiBN's parameters, y is the task (video recognition here) ground-truth, y^{det} is the ground-truth of the video types for training the adversarial video detector, $\mathcal{L}$ is the usual cross-entropy loss, and λ is a trade-off hyperparameter. The objectives $\mathcal{L}(x, y; \theta)$ and $\mathcal{L}(x, y^{det}; \theta^{det})$ are trained for clean data, while $\mathcal{L}(x + \delta_i, y; \theta)$ and $\mathcal{L}(x + \delta_i, y^{det}; \theta^{det})$ are for AT. $\mathcal{L}(x, y^{det}; \theta^{det})$ and $\mathcal{L}(x + \delta_i, y^{det}; \theta^{det})$ exclusively learns the BN selection module $h(\,)$, while $\mathcal{L}(x, y; \theta)$ and $\mathcal{L}(x + \delta_i, y; \theta)$ learns the entire framework $f(\,)$ in an end-to-end manner.

6.3.4 Defense Mechanism Against Unforeseen Attacks

In Sec. 6.3.1 and Sec.6.3.2, we mention how the proposed method addresses the presence of unforeseen attacks during inference. Here we further elaborate on its mechanism.

We deal with unforeseen attacks via proper attack type categorization, which is achieved by the proposed multiple BN structure and BN selection module. We consider PGD, ROA, AF and SPA attacks in this chapter. Suppose that we are aware of only PGD and ROA at training time. We classify PGD and ROA to the L_p-bounded attack group and the physically realizable attack group, respectively, based on the perturbation distributions. The MultiBN framework is built according

to this categorization. Specifically, we deploy three BN branches for clean data, L_p-bounded attacks and physically realizable attacks, respectively (see Figure 6.2). Each BN is adversarially trained to be robust against each particular category, and the BN selection module is trained to identify the most similar category of given input data.

Suppose that AF and SPA are unforeseen to us at training time but present at test time. During inference, the BN selection module identifies the most similar category of the input AF and SPA examples (i.e., the physically realizable attack group and the L_p-bounded attack group, respectively). Accordingly, the features from their belonging BN branch would dominate the feature maps after the feature aggregation step described in Eq. (6.11). Since each BN is robust against a particular attack group, MultiBN can achieve high robustness against the unforeseen AF and SPA attacks, which are classified as the most similar group by the BN selection module.

In contrast, a model without the multiple BN structure cannot be uniformly robust against multiple attack groups (see Sec. 6.4.2), resulting in sub-optimal robustness against unforeseen attacks. Besides, a model without the BN selection module cannot properly aggregate the features from different BN branches. Therefore, the proposed MultiBN framework consisting of the multiple BN structure and the BN selection module can decently address unforeseen attacks.

6.4 Experiments

In this section, we first describe our experimental setup. Second, we evaluate MultiBN's manually-controlled version to validate the effectiveness of the multiple BN structure and explore the properties of this structure. Next, we test the proposed

MultiBN's robustness and performance and compare it with state-of-the-art multi-perturbation training approaches. We also evaluate MultiBN's robustness against adaptive attacks, different attack budgets, and black-box attacks. Finally, we conduct further analyses on model size, sanity checks, and experiments on images.

6.4.1 Experimental Setup

Datasets. We use UCF-101 [198] and HMDB-51 [105] for evaluation, which are widely used video datasets in action recognition. UCF-101 consists of 13,320 videos from 101 action classes, and HMDB-51 has 6,766 videos from 51 action classes. Following [228], we resize their frame dimensions to 112×112 and uniformly sample each video into 40 frames. UCF-101 is the default dataset if not otherwise specified.

Attack setting. We consider L_∞-norm PGD [150], ROA [231], AF [253], and the proposed SPA attack. For PGD, we set the perturbation size ϵ to $4/255$; for ROA, we set the rectangle size s_{ROA} to 30×30 and ϵ to $255/255$; for AF, we set the framing width s_{AF} to 10 and ϵ to $255/255$; for SPA, we set the number of adversarial pixels on each frame s_{SPA} to 100 and ϵ to $255/255$. The number of attack iterations t_{max} is set to 5 for all the four attacks. To test the proposed method by strong attacks, all of these attacks are untargeted attacks and in the white-box setting (i.e., the attacker has full knowledge of the target model, including the multiple BN structure and the BN selection module).

Implementation details. We choose 3D ResNext-101 and 3D Wide ResNet-50 [65] as our target models, as they are two of the top-performing 3D CNNs for video recognition, where 3D ResNext-101 is the default target model if not otherwise

Table 6.1: Results (%) of MultiBN-manual on target model 3D ResNeXt-101 and dataset UCF-101. No Defense is trained on only clean data. AT-PGD, AT-ROA, AT-AF and AT-SPA are adversarially trained on a single specific attack type. The best results are in bold, and the best results among adversarially trained models are underlined.

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**89.0**	3.3	0.5	1.6	8.4	20.6	0.0
AT-PGD	78.6	**49.0**	5.0	0.6	67.1	40.1	0.3
AT-ROA	82.6	12.5	**69.0**	54.0	17.6	47.1	7.9
AT-AF	84.6	7.1	3.9	**80.5**	12.2	37.7	2.1
AT-SPA	83.5	36.9	2.6	0.7	**69.5**	38.6	0.2
MultiBN-manual	83.7	46.4	65.6	57.0	60.4	**62.6**	**40.7**

specified. For our adversarial video detector, we choose the lightweight 3D ResNet-18. We use the pre-trained weights from [65] and conduct AT upon the pre-trained models. We set MultiBN's Softmax temperature $\tau = 1$ and the trade-off hyperparameter $\lambda = 0.1$. All the models are trained by a SGD optimizer with initial learning rate $5e^{-4}$, momentum 0.9 and weight decay $1e^{-5}$, where the learning rate is decreased by a factor of 10 in the middle of the training process.

We apply the mean accuracy and the union accuracy as the metrics to evaluate the multi-perturbation robustness. The union accuracy requires that the target models correctly classify an input sample under all the considered input types.

6.4.2 Multiple Batch Normalization Structure

We first manually select the correct BN branches to investigate the effectiveness of the multiple BN structure. We call this variant MultiBN-manual. Then, we compare MultiBN-manual with vanilla AT [150] that trains on a single attack type. Table 6.1 shows that models trained on a specific attack always have the best robustness to that attack. AT-PGD and AT-ROA also yield high robustness to another attack in their

Table 6.2: Results (%) of each BN branch on the five input types. BN-Clean, BN-L_p and BN-Physical are the clean, PGD and ROA BN branches in the multiple BN structure, respectively.

BN Branch	Clean	PGD	ROA	AF	SPA
BN-Clean	**83.7**	21.3	13.5	5.9	23.8
BN-L_p	79.0	**46.4**	7.7	1.9	**60.4**
BN-Physical	83.0	23.5	**65.6**	**57.0**	26.6

own group, showing better generalization. However, all of them almost fail to defend against the attacks from other groups.

MultiBN-manual uniformly achieves the second-highest accuracy across all the five input types, and most of these accuracies are close to the best one. Although MultiBN-manual is not the best from the perspective of any specific input type, it sustains a much better balance when multiple input types are considered. This shows the effectiveness of the multiple BN structure in multi-perturbation robustness. As can be seen, MultiBN-manual's mean accuracy and union accuracy are significantly higher than AT on a single attack.

6.4.3 Analysis of Different Batch Normalization Branches

In the previous subsection, the attacker generates perturbations through the BN branch corresponding to its type. During inference, the input is also sent to that BN branch accordingly. In this subsection, we investigate the cases that perturbations are produced on a BN branch that is different from the group of the used attack type, and the input is sent to that BN branch during inference.

In Table 6.2, we can see that each BN branch performs the best on the input type on which they are trained on. Moreover, for unforeseen attacks, BN-L_p is the most robust to SPA, and BN-Physical is the most robust to AF. This result is consistent with

Table 6.3: Results (%) of the cases that the target BN and the inference BN are different.

Attack		PGD			ROA	
Inference BN \ Target BN	BN-Clean	BN-L_p	BN-Physical	BN-Clean	BN-L_p	BN-Physical
BN-Clean	21.3	50.9	35.9	13.5	17.9	56.6
BN-L_p	**72.6**	46.4	**70.5**	30.4	7.7	48.2
BN-Physical	46.4	**52.3**	23.5	**78.5**	**76.4**	**65.6**

Attack		AF			SPA	
Inference BN \ Target BN	BN-Clean	BN-L_p	BN-Physical	BN-Clean	BN-L_p	BN-Physical
BN-Clean	5.9	7.3	49.0	23.8	55.7	41.8
BN-L_p	16.4	1.9	33.5	**77.2**	**60.4**	**75.8**
BN-Physical	**75.2**	**62.5**	**57.0**	49.8	57.4	26.6

our assumption that feeding an unforeseen adversarial example to the BN branch of the same or the most similar group can enjoy the best benefit. Our grouping follows the observation that PGD and SPA have similar distributions (L_p-bounded attacks), and ROA and AF have similar distributions (physically realizable attacks).

6.4.4 Analysis of Target and Inference Batch Normalization

We further delve into the cases where an adversarial example is made inference on a BN branch (inference BN) different from the BN branch that is used to generate the adversarial example (target BN). In other words, we consider the cases that the target BN and the inference BN to be different.

The results in Table 6.3 are mostly consistent with that in Table 6.2, in which BN-L_p has the strongest robustness to L_p-bounded attacks, and BN-Physical has the strongest robustness to physically realizable attacks. PGD attack is an exception: When the target BN is BN-L_p, inference BN-L_p performs the worst.

In addition, we observe that for any specific inference BN, it is more robust to the adversarial examples generated on another BN branch, i.e., target BN and inference BN are different. In such a case, the attack is not a rigorous white-box attack, so we

Table 6.4: Results (%) of MultiBN and state-of-the-art approaches on target model 3D ResNeXt-101 and dataset UCF-101. The best results are in bold.

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**89.0**	3.3	0.5	1.6	8.4	20.6	0.0
TRADE [256]	82.3	29.0	5.7	3.3	42.2	32.5	1.9
AVG [205]	68.9	38.1	51.4	18.5	49.6	45.3	17.3
MAX [205]	72.8	32.5	31.0	5.8	49.4	38.3	5.5
MSD [153]	70.2	43.2	1.7	1.6	**56.0**	34.6	0.7
MultiBN (Ours)	74.2	**44.6**	**58.6**	**44.3**	53.7	**55.1**	**34.8**

Table 6.5: Results (%) of MultiBN and state-of-the-art approaches on target model 3D Wide ResNet-50 and dataset UCF-101. The best results are in bold.

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**88.4**	11.5	0.2	1.0	10.0	22.2	0.0
TRADE [256]	81.1	26.7	1.1	0.7	39.2	29.8	0.1
AVG [205]	74.5	43.1	55.6	3.5	57.2	46.8	3.5
MAX [205]	76.0	32.5	12.2	2.3	39.2	32.4	1.9
MSD [153]	71.0	46.3	2.9	0.9	**61.1**	36.4	0.2
MultiBN (Ours)	77.4	**46.5**	**59.9**	**48.1**	56.7	**57.7**	**37.8**

treat it as a kind of gray-box attack, in which the attacker does not know which BN branch would the adversarial example pass through during inference. This unveils that the attacks cannot perfectly transfer to other BN branches though the rest of the model parameters are shared in the same network. Such results show that the multiple BN structure can make secure against this gray-box setting.

6.4.5 Robustness of the Entire MultiBN Framework

In this subsection, we evaluate the entire MultiBN and compare it with state-of-the-art AT and multi-perturbation training approaches, including TRADES [256], AVG [205], MAX [205] and MSD [153]. For TRADES, we apply the AVG strategy to it for multi-perturbation training. Because we take clean data accuracy into consideration,

Table 6.6: Results (%) of MultiBN and state-of-the-art approaches on target model 3D ResNeXt-101 and dataset HMDB-51. The best results are in bold.

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**65.1**	0.0	0.0	0.0	0.3	13.1	0.0
TRADE [256]	54.8	6.8	0.3	0.0	20.5	16.5	0.0
AVG [205]	39.0	14.3	17.1	2.8	26.2	19.9	1.4
MAX [205]	48.6	13.9	16.0	0.1	30.3	21.8	0.0
MSD [153]	41.4	18.2	0.1	0.0	**31.2**	18.2	0.0
MultiBN (Ours)	51.1	**22.0**	**23.7**	**7.8**	29.9	**26.9**	**5.0**

we adjust AVG, MAX and MSD by involving clean data in training. That is, we add the clean data loss term $\mathcal{L}(x, y; \theta)$ into the expectation of objective functions Eq. (6.5), Eq. (6.6) and Eq. (6.8).

Table 6.4 reports the results on target model 3D ResNext-101 and dataset UCF-101, Table 6.5 reports the results on target model 3D Wide ResNet-50 and dataset UCF-101, Table 6.6 reports the results on target model 3D ResNext-101 and dataset HMDB-51. As expected, No Defense still achieves the best performance on clean data, showing that AT degrades clean data performance. TRADES has the best clean data performance among the AT approaches, but it lacks multi-perturbation robustness. AVG improves multi-perturbation robustness to a large extent, yet its clean data performance is very low. MAX is less robust than AVG in our case. MSD has the best and the second-best robustness against SPA and PGD, respectively, but it is vulnerable to physically realizable attacks. The proposed MultiBN achieves the second-best clean data performance among the defenses, the second-best robustness against SPA, and the best robustness against the rest of the attack types. MultiBN consistently outperforms the competitors in terms of mean accuracy and union accuracy by a wide margin, showing great multi-perturbation robustness. This holds true on different datasets and target models.

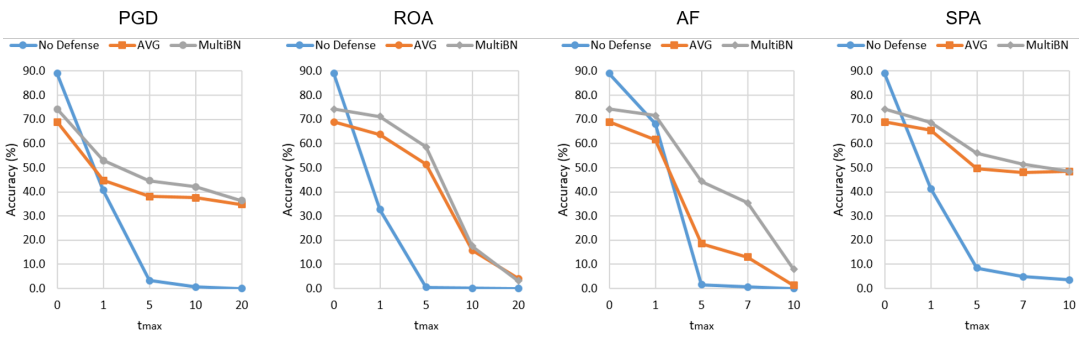

Figure 6.3: Results (%) under the four attack types with varied numbers of attack iterations.

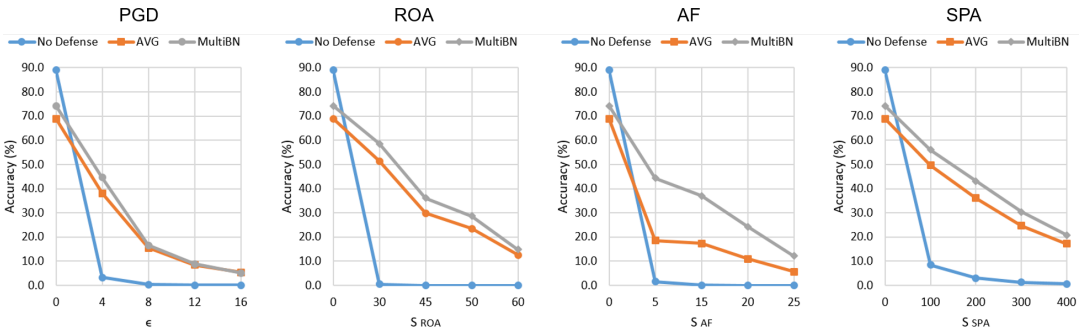

Figure 6.4: Results (%) under the four attack types with varied perturbation bounds.

6.4.6 Attack Budgets

To further evaluate the effectiveness of MultiBN, we test its scalability to different attack budgets. We vary the attack budgets by two aspects: The number of attack iterations t_{max} and the perturbation bounds of different attack types, i.e., PGD's ϵ, ROA's s_{ROA}, AF's s_{AF} and SPA's s_{SPA}. The results are presented in Figure 6.3 and Figure 6.4. The No Defense baseline and the strongest competitor AVG are compared.

We can see that MultiBN consistently achieves better robustness against different attack types with various attack iterations and perturbation bounds. This shows that MultiBN's multi-perturbation robustness is scalable to various attack budgets.

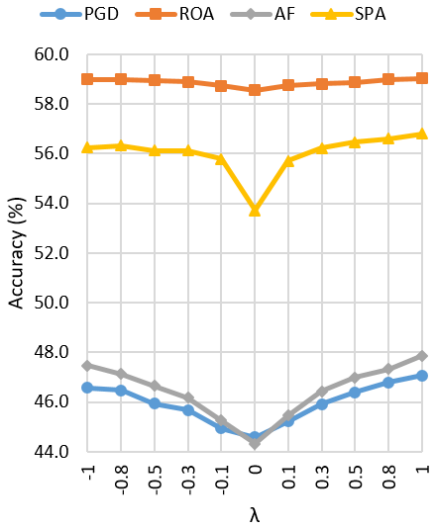

Figure 6.5: Results (%) of MultiBN under the adaptive attacks with varied λ of the four attack types.

6.4.7 Robustness Against Adaptive Attacks

To thoroughly evaluate MultiBN, we construct an adaptive attack [206], which jointly attacks the target model part and the BN selection module part. The intuition is to generate adversarial examples which can also fool the BN selection module to let it select the incorrect BN branch, and thus become easier to fool the target model. This adaptive attack is formulated as follows:

$$\delta = \arg\max_{\delta \in \mathrm{S}} \left[\mathcal{L}(x+\delta, y; \theta) + \lambda \cdot \mathcal{L}(x+\delta, y^{det}; \theta^{det}) \right]. \tag{6.14}$$

As presented in Figure 6.5, the canonical attack has the greatest attacking strength. The accuracies under all the four attack types monotonously increase as $|\lambda|$ increases. This shows that the considered adaptive attack fails to break MultiBN.

Table 6.7: Results (%) of MultiBN and state-of-the-art approaches under black-box attacks on UCF-101. The substitute model is the naturally trained 3D Wide ResNet-50, and the target model is 3D ResNeXt-101. The best results are shown in bold, and the second-best results are underlined.

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
TRADE [256]	**82.3**	**81.0**	60.8	<u>65.0</u>	**78.0**	**73.4**	49.3
AVG [205]	68.9	68.4	68.0	62.0	68.4	67.1	56.2
MAX [205]	72.8	72.4	<u>71.4</u>	63.5	<u>71.9</u>	70.4	<u>57.9</u>
MSD [153]	70.2	69.8	40.1	52.2	69.1	66.5	31.3
MultiBN (Ours)	<u>74.2</u>	<u>73.6</u>	**74.0**	**72.4**	71.5	<u>73.1</u>	**63.5**

6.4.8 Robustness Against Black-Box Attacks

In addition to the white-box robustness we discussed, we also evaluate the proposed method's robustness against black-box attacks [164]. Table 6.7 reports the results on UCF-101. Here we consider a naturally trained (i.e., train with only clean data) 3D Wide ResNet-50 as a substitute model to generate black-box adversarial examples, and test on the target model, 3D ResNeXt-101. As we can see, the proposed MultiBN uniformly achieves excellent robustness against multiple attack types in the black-box setting. In particular, MultiBN's robust accuracies are very close to its clean accuracy (74.2%), showing that the black-box attacks hardly fool it. Its union accuracy attains 63.5%, which significantly outperforms all the competitors.

6.4.9 Model Size Analysis

Apart from performance and robustness, model size is another critical factor when we evaluate a model. This regards the feasibility of a model for real-world applications. Our MultiBN significantly improves multi-perturbation robustness with only a minor increase in the number of parameters. To present the compactness of the MultiBN

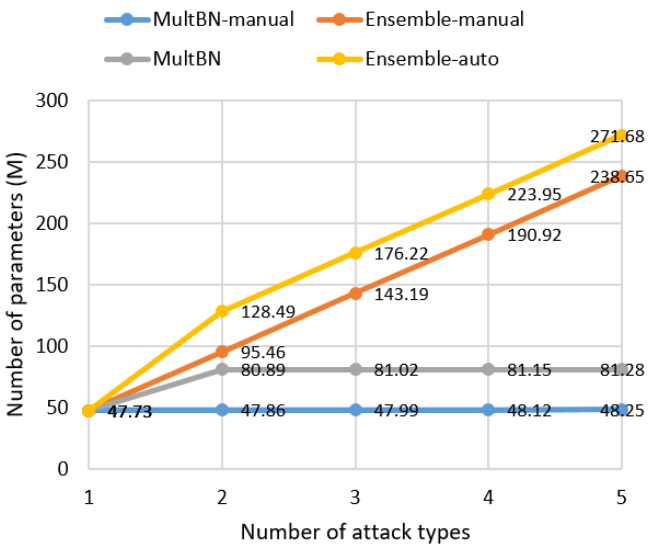

Figure 6.6: Model size analysis result. "-manual" refers to the model without the BN selection module.

architecture, we construct a naive model ensemble approach as a baseline for comparison. The model ensemble approach trains an individual model for each particular attack type, and uses our BN selection module to select the corresponding model for the input video during inference. Figure 6.6 compares the number of parameters of MultiBN and the model ensemble. The model ensemble's number of parameters linearly increases along with the number of attack types since its number of individual models equals the number of attack types. In contrast, MultiBN only deploys distinct BN parameters for each particular attack type and shares all the rest of the parameters across all the attack types. Hence, the increase of model size is minimal, especially compared to the backbone network's size. This demonstrates that the proposed method obtains excellent effectiveness and model compactness simultaneously.

Table 6.8: Results (%) of MultiBN and state-of-the-art approaches on target model ResNet-18 and dataset CIFAR-10. The best results are shown in bold, and the second-best results are underlined.

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**94.3**	0.0	4.7	0.1	16.3	23.1	0.0
TRADE [256]	71.4	14.7	34.7	30.4	52.8	40.8	10.1
AVG [205]	86.4	47.2	53.6	60.5	<u>67.8</u>	63.1	28.1
MAX [205]	87.7	46.3	60.0	54.6	**73.6**	<u>64.4</u>	<u>33.7</u>
MSD [153]	93.0	**52.7**	6.7	7.1	59.6	43.8	2.2
MultiBN (Ours)	<u>94.2</u>	<u>49.7</u>	**74.9**	**66.7**	60.9	**69.3**	**36.9**

6.4.10 Results on Images

The proposed method is effective in the image domain as well. For the experiment on images, we use CIFAR-10 [103] as the dataset and ResNet-18 [69] as the target model. The architecture of the adversarial video detector is also ResNet-18. Regarding attack setting, we set the perturbation size ϵ to $8/255$ for PGD, the rectangle size s_{ROA} to 12×12 for ROA, the framing width s_{AF} to 3 for AF, and the number of adversarial pixels on each image s_{SPA} to 30 for SPA. All the attacks are untargeted attacks and in the white-box setting.

Table 6.8 reports the evaluation results. Compared to the state-of-the-art approaches, MultiBN achieves the best accuracy under the ROA and AF attacks and the second-best accuracy under clean images and the PGD attack. Similar to the results in videos, MultiBN is far superior to all the competitors in terms of mean accuracy and union accuracy. This demonstrates that MultiBN can be a preferred solution for multi-perturbation robustness in both the image and the video domains.

6.4.11 Sanity Checks to Evaluation

To verify whether the proposed MultiBN's robustness is not due to obfuscated gradients, we report our results on the basic sanity checks introduced by [5]:

- Figure 6.3 shows that iterative attacks are stronger than one-step attacks.
- Table 6.4 and Table 6.7 show that white-box attacks are stronger than black-box attacks.
- Unbounded attacks reach 100.0% attack success rate (accuracy drops to 0.0%).
- Figure 6.4 shows that increasing distortion bound increases attack success (decreases accuracy).

These results confirm that our MultiBN's robustness is indeed not due to obfuscated gradients, which further demonstrates its reliability.

6.5 Summary

In this chapter, we proposed MultiBN, a new adversarial defense method aiming at multi-perturbation robustness. This is one of the first defenses against multiple and unforeseen adversarial videos. MultiBN uses a multiple BN structure to solve the distribution mismatch problem during multi-perturbation training. A BN selection module makes the entire framework automatic at inference time and differentiable for end-to-end training. Compared to existing AT approaches, MultiBN achieves stronger multi-perturbation robustness against different and even unforeseen L_p-bounded attacks and physically realizable attacks. This holds true on different datasets and target models. Furthermore, we conduct an extensive analysis to explore the

properties of the multiple BN structure under various conditions. In our future work, we will consider video-specific properties, such as temporal information, for adversarial attacks and defenses in videos.

Chapter 7

Adversarially Robust One-Class Novelty Detection

7.1 Motivation

One-class novelty detection refers to the problem of determining if a test data sample is normal (known class) or anomalous (novel class). In real-world applications, novel data is difficult to collect since they are often rare or unsafe. Hence, one-class novelty detection considers training data from only a single known class. Most recent advances in one-class novelty detection are based on the deep Auto-Encoder (AE) style architectures, such as Denoising Auto-Encoder (DAE) [186, 218], Variational Auto-Encoder (VAE) [102], Adversarial Auto-Encoder (AAE) [154, 170], Generative Adversarial Network (GAN) [59, 168, 180, 187, 258], etc. Given an AE that learns the distribution of the known class, normal data are expected to be reconstructed accurately, while anomalous data are not. The reconstruction error of the AE is then used as a score for a test example to perform novelty detection. Although deep novelty detection methods achieve impressive performance, their robustness against adversarial attacks [60, 201] lacks exploration.

Over the past few years, many adversarial attack and defense approaches have been proposed for tasks such as image classification [64, 173, 239, 244], video recognition (Chapter 3, 4, 6, and [228, 233]), optical flow estimation [175] and open-set recognition [190]. However, adversarial attacks or defenses have not been thoroughly investigated in the context of one-class novelty detection. We first show that present novelty detectors are vulnerable to adversarial attacks. Subsequently, we demonstrate that many state-of-the-art defenses [72, 193, 237, 239] prove to be sub-optimal to properly defend novelty detectors against adversarial examples. This motivates us to design an effective defense strategy specifically for one-class novelty detection.

To this end, we propose to leverage task-specific knowledge to protect novelty detectors. These novelty detectors are only required to retain information about normal data, thereby resulting in poor reconstructions for anomalous data. This is favorable to the novelty detection problem. This can be achieved by constraining the latent space to make the features closer to a prior distribution [165, 168]. Furthermore, it has been shown that adversarial perturbations can be removed in the feature space [239]. Therefore, one can largely manipulate the latent space of novelty detectors to devoid them of feature corruption introduced by adversaries, while maintaining the performance on clean input data. This property is unique to the novelty detection task, as most deep learning applications (e.g., image classification) require a model containing sophisticated semantic information, and a large manipulation on the latent space may limit the model capability, resulting in performance degradation.

In this chapter, we propose a defense strategy, referred to as Principal Latent Space (PrincipaLS), to defend novelty detectors against adversarial examples. Specifically, PrincipaLS learns the incrementally-trained [179] cascade principal components in the latent space. This contains a cascade Principal Component Analysis (PCA), which consists of a PCA operating on the vector dimension (i.e., channel) of a latent space [215] and the other PCA operating on the spatial dimension. We name these two PCAs as *Vector-PCA* and *Spatial-PCA*, respectively. First, Vector-PCA uses a learned *principal latent vector* to represent a latent space as the Vector-PCA space of a single-channel map. Since the principal latent vector is a pre-trained component that would not be affected by adversarial perturbations, most adversaries are removed at this step, and the remaining adversaries are enclosed within the small *Vector-PCA space*. Subsequently, Spatial-PCA uses learned *principal Vector-PCA maps* to represent the

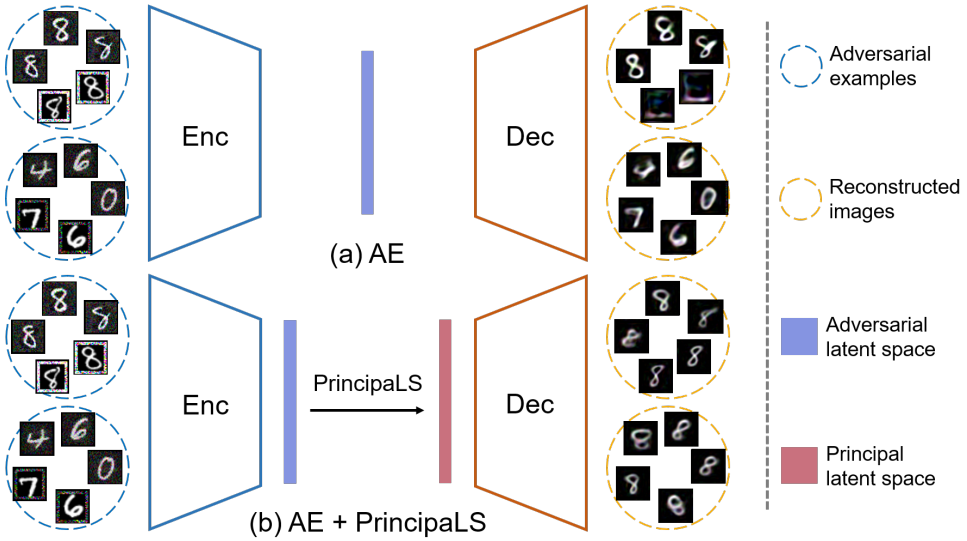

Figure 7.1: Overview of the proposed adversarially robust one-class novelty detection idea (PrincipaLS). The vanilla Auto-Encoder (AE) and AE+PrincipaLS are trained with the known class defined as digit 8. AE+PrincipaLS reconstructs every adversarial data into the known class (digit 8) and thus produces preferred reconstruction errors for novelty detection, even under attacks.

Vector-PCA space as the *Spatial-PCA space* and expel the remaining adversaries. Finally, the corresponding cascade inverse PCA transforms the Spatial-PCA space back to the original dimensionality, resulting in the *principal latent space*.

With PrincipaLS, the decoder could compute preferred reconstruction errors as novelty scores, even under adversarial attacks (see Figure 7.1). Additionally, we incorporate AT [150] with PrincipaLS to further exert PrincipaLS's ability in enhancing adversarial robustness. In contrast to typical defenses which often sacrifice their performance on clean data [209, 236], the proposed defense strategy does not hurt the performance but rather improves it. The PrincipaLS module can be attached to any AE-style architectures (VAE, GAN, etc.), so it can be applied to a wide variety of the existing novelty detection approaches, such as [102, 154, 170, 180, 186] etc. Moreover, the PrincipaLS module is lightweight and computationally efficient.

We establish a solid evaluation benchmark for the problem of adversarially robust one-class novelty detection. We extensively evaluate PrincipaLS on **eight** adversarial attacks (ranging from digital to physically realizable attacks and from white-box to black-box attacks), **five** datasets (ranging from toy to realistic datasets and from image to video datasets) and **seven** different novelty detectors. We further compare PrincipaLS with commonly-used defense methods and show that it consistently enhances the adversarial robustness of novelty detectors by significant margins. To the best of our knowledge, this is one of the first adversarially robust novelty detection methods. We hope that the provided evaluation benchmark and comprehensive baseline results for this emerging problem will be useful to the vision and machine learning communities. Code is available at: https://github.com/shaoyuanlo/PrincipaLS

The main contributions of this chapter are summarized as follows:

- We propose a novel adversarial defense method, PrincipaLS, based on task-specific knowledge to protect novelty detectors. To the best of our knowledge, this is one of the first adversarially robust novelty detection methods.
- We establish a solid evaluation benchmark for the problem of adversarially robust novelty detection.
- The proposed PrincipaLS consistently enhances the adversarial robustness of novelty detectors by wide margins. This holds true on multiple attacks, datasets and novelty detectors.
- We provide extensive analysis and discussion to study the proposed method and this emerging problem.
- We provide comprehensive baseline results for this emerging problem. These

baselines broadly cover eight adversarial attacks, five datasets and seven different novelty detectors.

7.1.1 Related Work

One-class novelty detection is of great interest to the computer vision community. Earlier algorithms mainly rely on Support Vector Machines (SVM) formulation [189, 202]. With the advent of deep learning, AE-based approaches are dominating this area and achieving state-of-the-art performance [58, 165, 168, 170, 180, 185, 186, 187, 232, 261]. ALOCC [180] considers a DAE [218] as a generator and appends a discriminator to train the entire network by the GAN framework [59]. GPND [170] is based on AAE [154], and it employs a discriminator to the latent space and the other discriminator to the output. OCGAN [168] includes two discriminators and a classifier to train a DAE by the GAN framework. ARAE [186] crafts adversarial examples from the latent space to adversarially train a DAE. Puzzle-AE [187] uses puzzle-solving as a pretext task to learn useful features, and it also incorporates adversarially robust training and the GAN training framework. Different from our work, ARAE and Puzzle-AE's adversarial examples aim to pursue performance, and their adversarial robustness is not thoroughly evaluated (see Sec. 7.5.1). To the best of our knowledge, APAE [61] might be the only present defense designed for anomaly detection. It uses approximate projection and feature weighting to reduce adversarial effects. However, its robustness is not fully tested and only anomalous data are perturbed in its evaluation (see Sec. 7.5.2). Instead, we provide a generic framework for evaluating the adversarial robustness of novelty detectors and our proposed defense method.

7.2 Attacking Novelty Detection Models

We consider several popular adversarial attacks [38, 60, 141, 150, 164, 253] (see details in Chapter 2) and modify their loss objectives to suit the novelty detection problem setup. Here we take PGD [150] as an example to illustrate our attack formulation. The other gradient-based attacks can be extended by a similar formulation.

Consider an AE-based target model with an encoder Enc and a decoder Dec, and an input image $\mathbf{X}$ with the ground-truth label $y \in \{-1, 1\}$, where "1" denotes the known class and "-1" denotes the novel classes. We generate the adversarial example $\mathbf{X}_{adv}$ as follows:

$$\mathbf{X}^{t+1} = Proj^{L_\infty}_{\mathbf{X},\epsilon}\{\mathbf{X}^t + \alpha \cdot sign(\bigtriangledown_{\mathbf{X}^t}\mathcal{L}(\hat{\mathbf{X}}^t, \mathbf{X}^t, y))\}, \tag{7.1}$$

where, $\hat{\mathbf{X}}^t = Dec(Enc(\mathbf{X}^t))$, $\alpha > 0$ denotes a step size, and $t \in [0, t_{max} - 1]$ is the number of attacking iterations, $\mathbf{X} = \mathbf{X}^0$ and $\mathbf{X}_{adv} = \mathbf{X}^{t_{max}}$. $Proj^{L_\infty}_{\mathbf{X},\epsilon}\{\cdot\}$ projects its element into an L_∞-norm bound with perturbation size ϵ such that $\| \mathbf{X}^{t+1} - \mathbf{X} \|_\infty \leq \epsilon$. $\mathcal{L}$ corresponds to the mean square error (MSE) loss defined as follows:

$$\mathcal{L}(\hat{\mathbf{X}}^t, \mathbf{X}^t, y) = y \| \hat{\mathbf{X}}^t - \mathbf{X}^t \|_2 . \tag{7.2}$$

Given a test example, if it belongs to the known class, we maximize its reconstruction error (i.e., novelty score) by gradient ascent; while if it belongs to novel classes, we minimize its reconstruction error by gradient descent. We use this formulation to generate adversarial examples for doing AT as well. During AT, since we can only access the training data of the known class, the label y is always 1 in Eq. (7.1) and Eq. (7.2).

Present novelty detection methods are vulnerable to these attacks (see Sec. 7.4.2);

that is, normal data would be misclassified into novel classes, and anomalous data would be misclassified into the known class. Moreover, this attacking strategy is much stronger than the attacks introduced by [186], which perturbs only normal data, and by [61], which perturbs only anomalous data. Because the proposed attack is stronger, our AT for defense is much more effective accordingly. A detailed comparison of the attacking strategies is discussed in Sec. 7.5.1 and Sec. 7.5.2. The proposed strong attack establishes a solid evaluation benchmark for the problem of adversarially robust one-class novelty detection.

7.3 Adversarially Robust Novelty Detection

The proposed defense strategy exploits the task-specific knowledge of one-class novelty detection. Specifically, we leverage the fact that a novelty detector's latent space can be manipulated to a larger extent as long as it retains the known class information. This property is especially useful to remove more adversarial perturbations in the latent space. Therefore, we propose to train a novelty detector by manipulating its latent space such that it can improve adversarial robustness while maintaining the performance on clean data. Note that these characteristics are specific to the novelty detection problem. The majority of visual recognition problems, such as image classification, require a model retaining multiple category information. Hence, a large manipulation on the latent space may hinder the model capability and thus degrade the performance. In the following subsections, we first briefly review PCA to define the notations used in this chapter, then discuss the proposed PrincipaLS in detail.

7.3.1 Preliminary

PCA computes the principal components of a collection of data and uses them to conduct a change of basis on the data through a linear transformation. Consider a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, its mean $\boldsymbol{\mu} \in \mathbb{R}^{1 \times d}$ and its covariance $\mathbf{C} = (\mathbf{X} - \boldsymbol{\mu})^\top (\mathbf{X} - \boldsymbol{\mu})$. $\mathbf{C}$ can be written as $\mathbf{C} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^\top$ via Singular Vector Decomposition (SVD), where $\mathbf{U} \in \mathbb{R}^{d \times d}$ is an orthogonal matrix containing the principal components of $\mathbf{X}$. Here we define a mapping h which computes the mean vector and the first k principal components of the given $\mathbf{X}$:

$$h(\mathbf{X}, k) : \mathbf{X} \rightarrow \{\boldsymbol{\mu}, \tilde{\mathbf{U}}\}, \tag{7.3}$$

where $\tilde{\mathbf{U}} \in \mathbb{R}^{d \times k}$ keeps only the first k columns of $\mathbf{U}$. Now we define the forward and the inverse PCA transformation as a pair of mapping ($f : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times k}$, $g : \mathbb{R}^{n \times k} \rightarrow \mathbb{R}^{n \times d}$); f performs the forward PCA:

$$f(\mathbf{X}; \boldsymbol{\mu}, \tilde{\mathbf{U}}) = (\mathbf{X} - \boldsymbol{\mu})\tilde{\mathbf{U}}, \tag{7.4}$$

and g performs the inverse PCA:

$$g(\mathbf{X}_{PCA}; \boldsymbol{\mu}, \tilde{\mathbf{U}}) = \mathbf{X}_{PCA}\tilde{\mathbf{U}}^\top + \boldsymbol{\mu}, \tag{7.5}$$

where $\mathbf{X}_{PCA} = f(\mathbf{X}; \boldsymbol{\mu}, \tilde{\mathbf{U}})$. Finally, we can write the PCA reconstruction of $\mathbf{X}$ as $\hat{\mathbf{X}} = g(f(\mathbf{X}; \boldsymbol{\mu}, \tilde{\mathbf{U}}); \boldsymbol{\mu}, \tilde{\mathbf{U}})$.

7.3.2 Principal Latent Space

The proposed PrincipaLS contains two major components: (1) Vector-PCA and (2) Spatial-PCA. In Vector-PCA, we perform (h, f, g) on the vector dimension as

(h_V, f_V, g_V), and in Spatial-PCA, we perform (h, f, g) on the spatial dimension as (h_S, f_S, g_S). Let Enc be the encoder and Dec be the decoder of a novelty detection model. Let us denote an adversarial image as $\mathbf{X}_{adv}$, we have its latent space $\mathbf{Z}_{adv} = Enc(\mathbf{X}_{adv}) \in \mathbb{R}^{s \times v}$, where $s = h \times w$ is the spatial dimensionality obtained by the product of height and width, and v is the vector dimensionality (i.e., the number of channels). Under adversarial attacks, $\mathbf{Z}_{adv}$ would be corrupted by adversarial perturbations such that the decoder cannot compute reconstruction errors favorable to novelty detection. We define the proposed PrincipaLS as a transformation $PrincipaLS : \mathbf{Z}_{adv} \rightarrow \mathbf{Z}_{PrincipaLS}$, which removes adversaries from $\mathbf{Z}_{adv}$, where $\mathbf{Z}_{PrincipaLS}$ is referred to as principal latent space. $PrincipaLS$ is implemented by our incrementally-trained cascade PCA. In the beginning, a sigmoid function replaces the encoder's last activation function to bound $\mathbf{Z}_{adv}$ values between 0 and 1. The following procedures are described below.

First, Vector-PCA computes the mean latent vector and the principal latent vector of $\mathbf{Z}_{adv}$:

$$\{\boldsymbol{\mu}_V, \tilde{\mathbf{U}}_V\} = h_V(\mathbf{Z}_{adv}, k_V = 1), \tag{7.6}$$

where, we always set k_V to 1, so $\tilde{\mathbf{U}}_V$ is the first principal latent vector of $\mathbf{Z}_{adv}$. Second, Vector-PCA transforms $\mathbf{Z}_{adv}$ to its Vector-PCA space $\mathbf{Z}_V \in \mathbb{R}^{s \times 1}$:

$$\mathbf{Z}_V = f_V(\mathbf{Z}_{adv}; \boldsymbol{\mu}_V, \tilde{\mathbf{U}}_V). \tag{7.7}$$

Next, Spatial-PCA computes the mean Vector-PCA map[1] and the principal Vector-PCA maps of $\mathbf{Z}_V$:

$$\{\boldsymbol{\mu}_S, \tilde{\mathbf{U}}_S\} = h_S(\mathbf{Z}_V^\top, k_S), \tag{7.8}$$

[1]We use the word "map" to indicate they are on the spatial dimension.

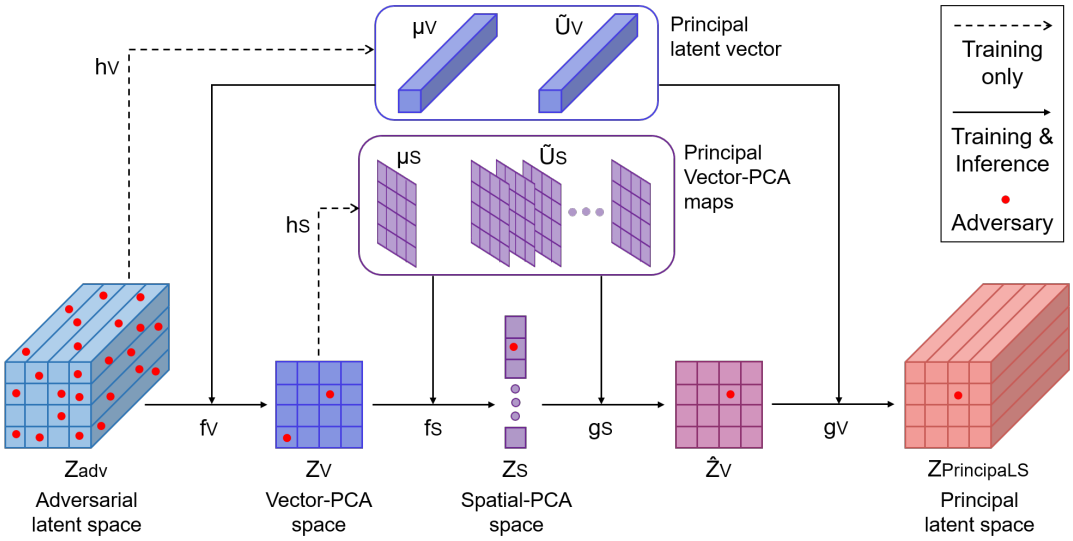

Figure 7.2: Overview of the proposed PrincipaLS. f_V: forward Vector-PCA, f_S: forward Spatial-PCA, g_S: inverse Spatial-PCA, g_V: inverse Vector-PCA, h_V and h_S are the mappings for computing principal components.

where k_S is a hyperparameter. Then, Spatial-PCA transforms $\mathbf{Z}_V$ to its Spatial-PCA space $\mathbf{Z}_S \in \mathbb{R}^{k_S \times 1}$:

$$\mathbf{Z}_S^\top = f_S(\mathbf{Z}_V^\top; \boldsymbol{\mu}_S, \tilde{\mathbf{U}}_S). \tag{7.9}$$

Finally, the inverse Spatial-PCA and the inverse Vector-PCA transform $\mathbf{Z}_S$ back to its original dimensionality:

$$\hat{\mathbf{Z}}_V^\top = g_S(\mathbf{Z}_S^\top; \boldsymbol{\mu}_S, \tilde{\mathbf{U}}_S), \tag{7.10}$$

$$\mathbf{Z}_{PrincipaLS} = g_V(\hat{\mathbf{Z}}_V; \boldsymbol{\mu}_V, \tilde{\mathbf{U}}_V), \tag{7.11}$$

where, $\hat{\mathbf{Z}}_V$ is the Spatial-PCA reconstruction of $\mathbf{Z}_V$, and $\mathbf{Z}_{PrincipaLS}$ is the resulting principal latent space. Figure 7.2 gives an overview of this procedure. The decoder then uses $\mathbf{Z}_{PrincipaLS}$ to reconstruct the input adversarial example as $\hat{\mathbf{X}}_{adv} = Dec(\mathbf{Z}_{PrincipaLS})$ for computing the novelty score.

7.3.3 Incremental Training

The *principal latent components* $\{\boldsymbol{\mu}_V, \tilde{\mathbf{U}}_V, \boldsymbol{\mu}_S, \tilde{\mathbf{U}}_S\}$ are incrementally-trained along with the network weights by Exponential Moving Average (EMA) during training, so we call this process incrementally-trained cascade PCA. Specifically, at training iteration t, these components are updated with the following equations:

$$\{\boldsymbol{\mu}_V^t, \tilde{\mathbf{U}}_V^t\} = (1-\eta_V)\{\boldsymbol{\mu}_V^{t-1}, \tilde{\mathbf{U}}_V^{t-1}\} + \eta_V \cdot h_V(\mathbf{Z}_{adv}^t), \tag{7.12}$$

$$\{\boldsymbol{\mu}_S^t, \tilde{\mathbf{U}}_S^t\} = (1-\eta_S)\{\boldsymbol{\mu}_S^{t-1}, \tilde{\mathbf{U}}_S^{t-1}\} + \eta_S \cdot h_S(\mathbf{Z}_V^{t\top}), \tag{7.13}$$

where η_V and η_S are the EMA learning rates.

Consider the model weights are trained by the mini-batch gradient descent with a batch size b, the latent dimensionality is shaped to $\mathbf{Z}_{adv} \in \mathbb{R}^{bs \times v}$, the resulting $\mathbf{Z}_V \in \mathbb{R}^{bs \times 1}$ is reshaped to $\mathbf{Z}_V \in \mathbb{R}^{s \times b}$ after the Vector-PCA f_V, and $\hat{\mathbf{Z}}_V \in \mathbb{R}^{s \times b}$ is reshaped back to $\hat{\mathbf{Z}}_V \in \mathbb{R}^{bs \times 1}$ after the inverse Spatial-PCA g_S. Hence, in a mini-batch, both h_V and h_S have b times more data points to acquire better principal latent components at each training iteration. At iteration t, (f_V, g_V) performs with the components $\{\boldsymbol{\mu}_V^t, \tilde{\mathbf{U}}_V^t\}$, and (f_S, g_S) performs with the components $\{\boldsymbol{\mu}_S^t, \tilde{\mathbf{U}}_S^t\}$. When the training process ends, the well-trained components are denoted as $\{\boldsymbol{\mu}_V^*, \tilde{\mathbf{U}}_V^*, \boldsymbol{\mu}_S^*, \tilde{\mathbf{U}}_S^*\}$. During infernce, (f_V, g_V) performs with $\{\boldsymbol{\mu}_V^*, \tilde{\mathbf{U}}_V^*\}$, and (f_S, g_S) performs with $\{\boldsymbol{\mu}_S^*, \tilde{\mathbf{U}}_S^*\}$, while h_V and h_S do not operate (see Figure 7.2). The entire process is differentiable during inference and thus does not cause obfuscated gradients [5].

This incremental training helps make sure that the cascade PCA is aware of the network weight updates at each training step, and vice versa [215]. Therefore, when one is updated, the other one would be updated accordingly. The incremental training

encourages mutual learning between the principle latent components and the network weights. The entire model and thus can be well-trained end-to-end.

7.3.4 Defense Mechanism

We further elaborate on how the proposed PrincipaLS defends against adversarial attacks. Given an adversarial example $\mathbf{X}_{adv}$, its latent space $\mathbf{Z}_{adv}$ is adversarially perturbed. After Vector-PCA, each latent vector of $\mathbf{Z}_{adv}$ is represented by a scaling factor of the learned principal latent vector $\tilde{\mathbf{U}}_V^*$ (with a bias term $\boldsymbol{\mu}_V^*$). The Vector-PCA space $\mathbf{Z}_V$ stores these scaling factors on a single-channel map (i.e., on the spatial domain only). Since all the principal latent components are pre-trained parameters, they would not be affected by adversarial perturbations. Replacing the perturbed latent vectors by $\tilde{\mathbf{U}}_V^*$ removes the majority of the adversaries. The only place where the remaining adversaries can appear is the scaling factors of $\tilde{\mathbf{U}}_V^*$ on the single-channel map. In other words, these adversaries are enclosed within a small subspace, making them easier to expel.

Subsequently, Spatial-PCA reconstructs this small subspace by a set of principal Vector-PCA maps $\tilde{\mathbf{U}}_S^*$ (with a bias term $\boldsymbol{\mu}_S^*$). Since $\tilde{\mathbf{U}}_S^*$ and $\boldsymbol{\mu}_S^*$ are adversary-free, the remaining adversaries are further removed. From another perspective, this step can be viewed as PCA-based denoising performed in the spatial domain of features. With the robust principal latent space $\mathbf{Z}_{PrincipaLS}$, the decoder can obtain a preferred reconstruction error for novelty detection, even in the presence of an adversarial example. Additionally, we perform AT [150] to train the model, further improving the robustness.

7.4 Experiments

We evaluate PrincipaLS on eight adversarial attacks, five datasets and seven existing novelty detection methods. We further compare PrincipaLS with state-of-the-art defense approaches. An extensive analysis is also presented.

7.4.1 Experimental Setup

Datasets. We use five datasets for evaluation: MNIST [116], Fashion-MNIST (F-MNIST) [235], CIFAR-10 [103], MVTec-AD [10] and ShanghaiTech (SHTech) [129]. MNIST consists of grayscale handwritten digits from 0 to 9. It contains 60,000 training data and 10,000 test data. F-MNIST is composed of grayscale images from 10 fashion product categories. It comprises 60,000 training data and 10,000 test data. CIFAR-10 consists of color images from 10 different classes. There are 50,000 training and 10,000 test images in this dataset. MVTec-AD is an anomaly detection dataset that consists of color images from 15 objects and textures categories. Each category contains normal and anomalous images with different types of defects. There are 3,629 training and 1,725 (467 normal and 1258 anomalous) test images in this dataset. SHTech is a video anomaly detection dataset that consists of videos from 13 scenes. It contains 274,515 training and 40,791 (23,465 normal and 17,326 anomalous) test frames. It is the largest dataset among existing anomaly detection benchmarks. In our experiments, we resize all the datasets to 32×32 during both training and testing.

Evaluation protocol. For the MNIST, F-MNIST and CIFAR-10 datasets, which are originally created for image classification, we simulate a one-class novelty detection scenario by the following protocol. Given a dataset, each class is defined as the known class at a time, and a model is trained with the training data of this known class.

During inference, the test data of the known class are considered normal, and the test data of the other classes (i.e., novel classes) are considered anomalous. We select the anomalous data from each novel class equally to constitute half of the test set, where the anomalous data within a novel class are selected randomly. Hence, our test set contains 50% anomalous data, where each novel class accounts for the same proportion. The area under the Receiver Operating Characteristic curve (AUROC) value is used as the evaluation metric, where the ROC curve is obtained by varying the threshold of the novelty score. For each dataset, we report the mean AUROC (mAUROC) across its 10 classes.

For the MVTec-AD dataset, we conduct experiments on all the 15 categories and report mAUROC across these 15 categories. Similarly, for each category, we sample the anomalous data from each defect type equally to constitute half of the test set such that the test set contains 50% anomalous data. For the SHTech dataset, we directly use its default test set as its normal-to-anomalous ratio is more balanced. Following [151, 180], we report frame-level AUROC.

Baseline defenses. To the best of our knowledge, APAE [61] might be the only present defense designed for anomaly detection. In addition to APAE, we implement five commonly-used defenses, which are originally designed for classification tasks, in the context of novelty detection. They are PGD-AT [150], FD [239], SAT [237], RotNet-AT [72] and SOAP [193], where FD, SAT and RotNet-AT incorporate PGD-AT. We use Gaussian non-local means [17] for FD, Swish [70] for SAT, and RotNet [56] for SOAP. These are their well-performing versions.

Benchmark novelty detectors. We apply PrincipaLS to seven novelty detection methods, including a vanilla AE, VAE [102], AAE [154], ALOCC [180], GPND

[170], ARAE [186] and Puzzle-AE [187], where the vanilla AE is the default novelty detector if not otherwise specified. PrincipaLS is added after the last layer of the novelty detection models' encoder.

In order to evenly evaluate the adversarial robustness of these approaches, we unify their AE backbones into the following architecture. The encoder consists of four 3×3 convolutional layers, where each of the first three layers is followed by a 2×2 max-pooling with stride 2. We use a base channel size of 64, and increase the number of channels by a factor of 2. The decoder mirrors the encoder but replaces every max-pooling by a bilinear interpolation with a factor of 2. All the convolutional layers are followed by a batch normalization layer [82] and ReLU.

Attack setting. We test adversarial robustness against five white-box attacks, including FGSM [60], PGD [150], MI-FGSM [38], MultAdv (proposed in Chapter 3) and AF [253], where PGD is the default attack if not otherwise specified. A black-box attack and two adaptive attacks [164, 206] are also considered. All the attacks are implemented based on the formulations in Sec. 7.2.

For FGSM, PGD and MI-FGSM, we set ϵ to $25/255$ for MNIST, $16/255$ for F-MNIST, $8/255$ for CIFAR-10, $2/255$ for MVTec-AD, and $8/255$ for SHTech. For MultAdv, we set ϵ_m to 1.25 for MNIST, 1.16 for F-MNIST, 1.08 for CIFAR-10, 1.02 for MVTec-AD, and 1.08 for SHTech. For AF, we set ϵ to $160/255$, $120/255$, $80/255$, $20/255$ and $80/255$ for MNIST, F-MNIST, CIFAR-10, MVTec-AD and SHTech, respectively. The framing width w_{AF} is set to 1. The number of attack iterations t_{max} is set to 1 for FGSM and 5 for the other attacks. All the defenses that incorporate PGD-AT (i.e., PGD-AT, FD, SAT, RotNet-AT and our PrincipaLS) use the PGD setting described here for doing AT.

Implementation details. We implement experiments by PyTorch [166]. All the models are trained by Adam optimizer [101] with initial learning rate $5e^{-5}$ and weight decay $1e^{-4}$ for 50 epochs (except that 10 epochs for SHTech), where the learning rate is decreased by a factor of 10 at the 20th and 40th epochs. The batch size is 128. For PrincipaLS, we set k_V to 1, k_S to 8, initial η_V to 0.1 and initial η_S to 0.001, where η_V and η_S are also decreased by a factor of 10 at the 20th and 40th epochs.

7.4.2 White-Box Robustness

The robustness of one-class novelty detection against various white-box attacks is reported in Table 7.1, where the vanilla AE is used. Without a defense, mAUROC scores drop significantly under all the white-box attacks, which shows the vulnerability of novelty detectors to adversarial examples. PGD-AT improves adversarial robustness to a great extent. FD makes a slight improvement upon PGD-AT in most cases. SAT and Rot-AT seem not effective upon PGD-AT in the context of novelty detection. SOAP performs well in some cases but not uniformly. Compared to other methods, APAE generally shows less robustness. The proposed method, PrincipaLS, significantly increases mAUROC with PGD-AT, leading the other defenses by a decent margin. Moreover, PrincipaLS is consistently better across all the five white-box attacks, ranging from digital attacks to physically realizable attacks; on five datasets, ranging from toy datasets to realistic datasets, and from the image domain to the video domain.

PrincipaLS-knowledgeable attacks. As discussed above, in a white-box attack, attackers are aware of the presence of the defense mechanism, i.e., PrincipaLS (it is differentiable at inference time, see Sec. 7.3). However, they count on only the

Table 7.1: The mAUROC of models under various adversarial attacks.

Dataset	Defense	Clean	FGSM	PGD	MI-FGSM	MultAdv	AF	Black-box	Mean
MNIST [116]	No Defense	0.964	0.350	0.051	0.022	0.170	0.014	0.790	0.337
	PGD-AT [150]	0.961	0.604	0.357	0.369	0.444	0.155	0.691	0.512
	FD [239]	0.963	0.612	0.366	0.379	0.453	0.142	0.700	0.516
	SAT [237]	0.947	0.527	0.295	0.306	0.370	0.142	0.652	0.463
	RotNet-AT [72]	0.967	0.598	0.333	0.333	0.424	0.101	0.695	0.493
	SOAP [193]	0.940	0.686	0.504	0.506	0.433	0.088	0.863	0.574
	APAE [61]	0.925	0.428	0.104	0.105	0.251	0.022	0.730	0.366
	PrincipaLS (Ours)	**0.973**	**0.812**	**0.706**	**0.707**	**0.725**	**0.636**	**0.866**	**0.775**
F-MNIST [235]	No Defense	0.892	0.469	0.088	0.047	0.148	0.112	0.562	0.331
	PGD-AT [150]	0.890	0.518	0.368	0.348	0.327	0.253	0.540	0.463
	FD [239]	0.886	0.524	0.379	0.359	0.335	0.252	0.535	0.467
	SAT [237]	0.878	0.444	0.306	0.285	0.273	0.231	0.492	0.416
	RotNet-AT [72]	0.891	0.527	0.375	0.351	0.312	0.240	0.541	0.462
	SOAP [193]	0.876	0.639	0.475	0.475	0.327	0.274	0.611	0.525
	APAE [61]	0.861	0.510	0.174	0.174	0.220	0.135	0.513	0.370
	PrincipaLS (Ours)	**0.909**	**0.687**	**0.613**	**0.599**	**0.590**	**0.605**	**0.711**	**0.673**
CIFAR-10 [103]	No Defense	0.550	0.186	0.034	0.018	0.025	0.035	0.227	0.154
	PGD-AT [150]	0.546	0.236	0.145	0.139	0.107	0.096	0.223	0.213
	FD [239]	0.546	0.237	0.147	0.141	0.109	0.103	0.222	0.215
	SAT [237]	0.537	0.223	0.141	0.135	0.101	0.079	0.219	0.205
	RotNet-AT [72]	0.547	0.236	0.139	0.107	0.075	0.092	0.224	0.203
	SOAP [193]	0.546	0.270	0.131	0.141	0.096	0.070	0.231	0.211
	APAE [61]	0.552	0.259	0.097	0.097	0.077	0.112	0.255	0.207
	PrincipaLS (Ours)	**0.577**	**0.320**	**0.246**	**0.243**	**0.202**	**0.244**	**0.333**	**0.309**
MVTec-AD [10]	No Defense	0.667	0.111	0.032	0.022	0.034	0.061	**0.595**	0.217
	PGD-AT [150]	0.655	0.123	0.053	0.040	0.054	0.062	0.569	0.222
	FD [239]	0.658	0.145	0.061	0.050	0.061	0.066	0.572	0.230
	SAT [237]	0.636	0.083	0.029	0.024	0.035	0.044	0.553	0.201
	RotNet-AT [72]	**0.677**	0.123	0.050	0.038	0.049	0.059	0.586	0.226
	SOAP [193]	0.540	0.167	0.092	0.056	0.095	**0.456**	0.582	0.284
	APAE [61]	0.621	0.142	0.058	0.044	0.058	0.120	0.553	0.228
	PrincipaLS (Ours)	0.638	**0.334**	**0.243**	**0.238**	**0.197**	0.164	0.542	**0.337**
SHTech [129]	No Defense	0.523	0.204	0.034	0.038	0.006	0.000	0.220	0.146
	PGD-AT [150]	0.527	0.217	0.168	0.154	0.100	0.000	0.221	0.198
	FD [239]	0.528	0.226	0.189	0.181	0.132	0.002	0.229	0.212
	SAT [237]	**0.529**	0.184	0.110	0.092	0.040	0.000	0.199	0.165
	RotNet-AT [72]	0.516	0.220	0.163	0.158	0.113	0.000	0.229	0.200
	SOAP [193]	0.432	0.024	0.002	0.000	0.002	**0.181**	0.202	0.120
	APAE [61]	0.510	0.215	0.048	0.050	0.011	0.000	0.207	0.149
	PrincipaLS (Ours)	0.498	**0.274**	**0.223**	**0.217**	**0.175**	0.051	**0.308**	**0.249**

novelty detection objective (i.e., MSE loss, see Eq. (7.2)) to generate adversarial examples. We follow the practice of the most recent adversarial defense studies such as [193], to thoroughly evaluate the proposed defense mechanism. More precisely,

we try to find an adaptive attack [164, 206] by giving the full knowledge of the PrincipaLS defense mechanism to the attacker. We refer to this type of attack as *PrincipaLS-knowledgeable attack*.

We construct two PrincipaLS-knowledgeable attacks, Knowledgeable A and Knowledgeable B, on top of the PGD attack. They jointly optimize Eq. (7.2) and an auxiliary loss developed with the knowledge of PrincipaLS. Knowledgeable A attempts to minimize the L_2-norm between the latent space before and after the PrincipaLS transformation. The intuition is to void PrincipaLS such that the input and the output latent space of PrincipaLS become closer. In other words, Knowledgeable A replaces Eq. (7.2) with the following equation:

$$\mathcal{L} = y \parallel \hat{\mathbf{X}}^t - \mathbf{X}^t \parallel_2 - \lambda_A \parallel \mathbf{Z}^t_{PrincipaLS} - \mathbf{Z}^t_{adv} \parallel_2, \tag{7.14}$$

where λ_A is a trade-off parameter. Knowledgeable B attempts to maximize the L_2-norm between the latent space of the current adversarial example $\mathbf{X}^t$ and its clean counterpart $\mathbf{X}^0$ after the PrincipaLS transformation. The intuition is to keep the adversarial latent space away from the clean one. In other words, Knowledgeable B replaces Eq. (7.2) with the following equation:

$$\mathcal{L} = y \parallel \hat{\mathbf{X}}^t - \mathbf{X}^t \parallel_2 + \lambda_B \parallel \mathbf{Z}^t_{PrincipaLS} - \mathbf{Z}^0_{PrincipaLS} \parallel_2, \tag{7.15}$$

where λ_B is a trade-off parameter. When $\lambda_A = 0$ or $\lambda_B = 0$, the PrincipaLS-knowledgeable attacks reduce to the conventional white-box attacks.

In Figure 7.3, we can observe that mAUROC monotonously increases as $|\lambda_A|$ or $|\lambda_B|$ increases. That is, these PrincipaLS-knowledgeable attacks cannot further reduce PrincipaLS's mAUROC, and the additional auxiliary loss terms would attenuate the

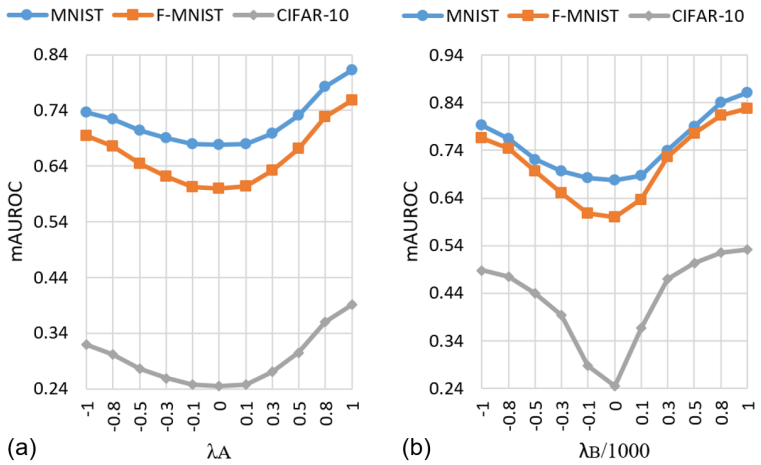

Figure 7.3: The mAUROC of PrincipaLS under PrincipaLS-knowledgeable attacks with varied trade-off parameters. (a) Knowledgeable A. (b) Knowledgeable B.

MSE loss gradients. This indicates that attackers cannot straightforwardly benefit from the knowledge of PrincipaLS. Hence, the conventional white-box attack still has the greatest attacking strength. This result shows that it is not easy to find a stronger attack to break PrincipaLS, even with the full knowledge of the PrincipaLS mechanism.

7.4.3 Black-Box Robustness

The robustness against black-box attacks [164] is shown in the second last column of Table 7.1. Here we consider a naturally trained (i.e., train with only clean data) GPND as a substitute model and apply MI-FGSM, which has better transferability, to generate black-box adversarial examples for target models. As we can see, the defenses with PGD-AT degrade black-box robustness, which is identical to the observation in classification tasks [207]. SOAP, which is without using AT, shows better black-box robustness. PrincipaLS greatly improves the black-box robustness on most datasets even with PGD-AT. The naturally trained PrincipaLS model achieves 0.907, 0.742

Table 7.2: The mAUROC of models under PGD attack. Various novelty detectors are used.

Dataset	Defense	Test type	AE	VAE	AAE	ALOCC	GPND	ARAE	Puzzle-AE
MNIST [116]	No Defense	Clean	0.964	0.979	0.973	0.961	0.946	0.965	0.967
	No Defense	PGD	0.051	0.087	0.056	0.141	0.128	0.133	0.295
	PGD-AT [150]	PGD	0.357	0.521	0.427	0.312	0.582	0.341	0.319
	FD [239]		0.366	0.525	0.419	0.319	0.551	0.350	0.322
	SAT [237]		0.295	0.485	0.470	0.330	0.527	0.254	0.286
	RotNet-AT [72]		0.333	0.501	0.507	0.361	0.551	0.314	0.315
	SOAP [193]		0.504	0.608	0.398	0.606	0.425	0.522	0.533
	APAE [61]		0.104	0.155	0.240	0.202	0.229	0.191	0.278
	PrincipaLS (Ours)		**0.706**	**0.739**	**0.608**	**0.693**	**0.741**	**0.695**	**0.599**
F-MNIST [235]	No Defense	Clean	0.892	0.914	0.912	0.901	0.915	0.901	0.911
	No Defense	PGD	0.088	0.223	0.152	0.177	0.177	0.262	0.438
	PGD-AT [150]	PGD	0.368	0.538	0.512	0.367	0.539	0.420	0.463
	FD [239]		0.379	0.533	0.513	0.370	0.542	0.428	0.470
	SAT [237]		0.306	0.504	0.499	0.332	0.530	0.351	0.410
	RotNet-AT [72]		0.375	0.542	0.509	0.365	0.524	0.396	0.429
	SOAP [193]		0.475	0.509	0.313	0.477	0.386	0.548	0.521
	APAE [61]		0.174	0.366	0.300	0.246	0.398	0.310	0.409
	PrincipaLS (Ours)		**0.613**	**0.604**	**0.599**	**0.612**	**0.626**	**0.599**	**0.629**
CIFAR-10 [103]	No Defense	Clean	0.550	0.552	0.555	0.551	0.559	0.578	0.544
	No Defense	PGD	0.034	0.073	0.051	0.037	0.027	0.087	0.141
	PGD-AT [150]	PGD	0.145	0.177	0.195	0.146	0.182	0.157	0.167
	FD [239]		0.147	0.180	0.206	0.150	0.187	0.152	0.170
	SAT [237]		0.141	0.170	0.186	0.141	0.181	0.107	0.160
	RotNet-AT [72]		0.139	0.163	0.161	0.105	0.147	0.101	0.132
	SOAP [193]		0.131	0.094	0.043	0.172	0.075	0.117	0.204
	APAE [61]		0.097	0.179	0.171	0.095	0.062	0.154	0.193
	PrincipaLS (Ours)		**0.246**	**0.247**	**0.252**	**0.244**	**0.242**	**0.245**	**0.248**

and 0.332 mAUROC on MNIST, F-MNIST and CIFAR-10, respectively, under the black-box attack.

7.4.4 Generalizability

Table 7.2 shows the adversarial robustness of various state-of-the-art novelty detection models. All of them are susceptible to adversarial attacks. We attach the PrincipaLS module to these models to protect them. We can see that PrincipaLS uniformly robustifies all of these novelty detectors and significantly outperforms the other defense approaches. This confirms that PrincipaLS can be applied to a wide variety of

Table 7.3: The mAUROC of models under clean data.

Defense	MNIST	F-MNIST	CIFAR-10
No Defense	0.964	0.892	0.550
FD [239]	0.965	0.892	0.551
SAT [237]	0.949	0.883	0.543
RotNet-AT [72]	0.963	0.897	0.554
SOAP [193]	0.940	0.876	0.546
APAE [61]	0.925	0.861	0.552
PrincipaLS (Ours)	**0.973**	**0.922**	**0.578**

the present novelty detection methods, demonstrating its excellent generalizability.

7.4.5 Performance on Clean Data

We also evaluate the performance of PrincipaLS on clean data. In this experiment, all the models are naturally trained. As shown in Table 7.3, PrincipaLS improves the performance upon the original network architecture (No Defense), while, the other defenses do not make obvious improvements. This shows that PrincipaLS generalizes better for both clean data and adversarial examples. PrincipaLS enjoys this benefit because the principal latent components are learned from only the latent space of the known class. Due to this, when transforming the latent space of any novel class image, PrincipaLS projects it into the known class space defined by the principal latent component. This brings the transformed latent space closer to the latent space of the known class, resulting in the decoder trying to reconstruct it into a known class image. Subsequently, this produces high reconstruction error for the novel class images while barely affecting the reconstruction of the known class images.

Table 7.4: The inference speed of each defense. The test images are from CIFAR-10 with an input size of 32 $\times$ 32. The experiment is performed on a single NVIDIA RTX 2080 Ti GPU.

Defense	Speed (FPS)	Difference
No Defense	18.0 $\times 10^3$	
FD [239]	6.8 $\times 10^3$	-62.2%
SAT [237]	18.2 $\times 10^3$	+1.1%
RotNet-AT [72]	18.0 $\times 10^3$	-0.0%
SOAP [193]	3.1 $\times 10^3$	-82.2%
APAE [61]	4.0 $\times 10^3$	-77.8%
PrincipaLS (Ours)	15.6 $\times 10^3$	-13.3%

Table 7.5: The mAUROC of different PrincipaLS variants under PGD attack.

Defense	MNIST	F-MNIST	CIFAR-10
PGD-AT [150]	0.357	0.368	0.145
Vector-PCA	0.566	0.499	0.215
Vector-PCA+FD	0.582	0.505	0.215
PrincipaLS (Ours)	**0.706**	**0.613**	**0.246**

7.4.6 Inference Speed

The PrincipaLS module is lightweight and computationally efficient. We test the inference speed of each defense via images from CIFAR-10 with an input size of 32 $\times$ 32. The experiment is performed on a single NVIDIA RTX 2080 Ti GPU. As can be seen in Table 7.4, when the PrincipaLS module is attached to an AE, the inference speed only decreases by 13.3%. This cost turns to significant improvements in robustness. We use a compact AE architecture as described in Sec. 7.4.1. If a deeper AE architecture is considered, PrincipaLS's relative computational overhead will be even lower. In contrast, FD contains a heavy feature denoising block which decreases inference speed by 62.2%. SOAP and APAE rely on adversarial purification processes at inference time, greatly increasing computational costs.

Table 7.6: The trade-off analysis of PrincipaLS's k_V and k_S values on MNIST dataset.

Input	Original AE	$k_V = 1$	$k_V = 4$	$k_V = 16$	$k_V = 64$
Clean	0.964	0.973	0.975	0.971	0.971
PGD	0.357	0.706	0.621	0.581	0.557
Input	Vec-PCA only	$k_S = 1$	$k_S = 4$	$k_S = 8$	$k_S = 12$
Clean	0.968	0.937	0.951	0.973	0.973
PGD	0.566	0.549	0.681	0.706	0.667

7.4.7 Analysis

Ablation study. Table 7.5 reports the results of different PrincipaLS variants. First, Vector-PCA alone significantly improves the robustness upon PGD-AT. This shows that the mechanism of replacing perturbed latent vectors by the incrementally-trained principal latent vector is effective. As discussed earlier, in PrincipaLS the adversaries can stay only on the scaling factors of the principal latent vector. Next, we further remove the adversaries with the help of denoising operation on the spatial dimension. We try to deploy a feature denoising block [239] after the forward Vector-PCA. This baseline is denoted as *Vector-PCA+FD*. This makes a slight improvement over the Vector-PCA baseline. Finally, the complete PrincipaLS uses Spatial-PCA for this purpose instead, achieving great mAUROC increase. This shows Spatial-PCA's advantage over FD in our case.

Trade-off of k_V and k_S values. We look into the trade-off of PrincipaLS's k_V and k_S values. Table 7.6 reports the results on the MNIST dataset. For both varying k_V (fix k_S=8) and k_S (fix k_V=1), we observe that larger k leads to lower PGD accuracy but higher clean accuracy in general. The reason is that using larger k retains more semantic information of feature maps while keeping more adversaries simultaneously. k_S=1 is an exception. It has lower PGD accuracy because it loses too much information.

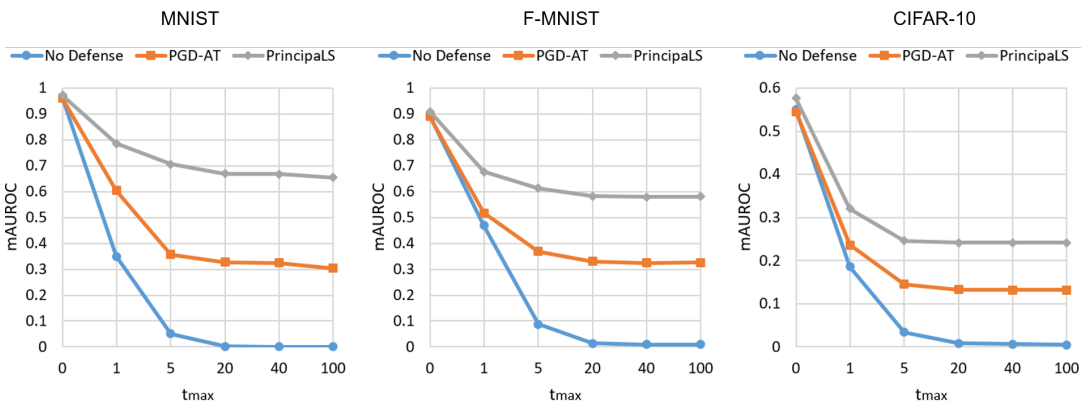

Figure 7.4: The mAUROC of models under PGD attack with varied numbers of attack iterations t_{max}.

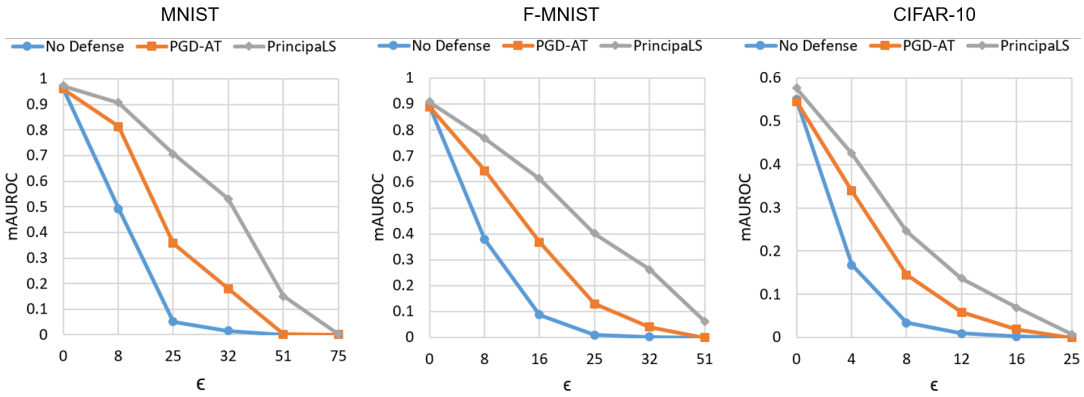

Figure 7.5: The mAUROC of models under PGD attack with varied perturbation sizes ϵ.

According to this trade-off analysis, we set k_V=1 and k_S=8 for PrincipaLS as discussed in Sec. 7.4.1.

Attack budgets. To fully evaluate the effectiveness of the proposed PrincipaLS, we test its scalability to different attack budgets. We vary the attack budgets by two aspects: The number of attack iterations t_{max} and perturbation size ϵ. The results are presented in Figure 7.4 and Figure 7.5, respectively.

First, we can see that the attack strength does not increase obviously along with the increase of t_{max}. This observation is consistent with that of Madry et al. [150] and

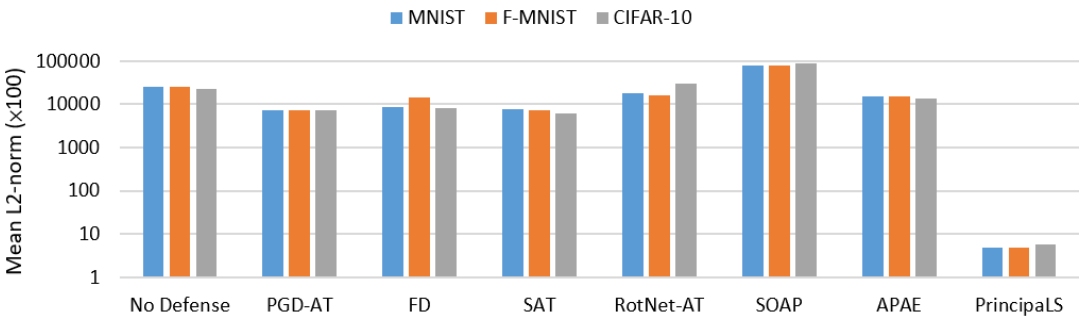

Figure 7.6: Mean L_2-norm between the latent space of PGD adversarial examples and that of their clean counterpart on different defenses. The values are the mean over an entire dataset.

Xie et al. [239]. The proposed PrincipaLS shows constant adversarial robustness and consistently performs better than No Defense and PGD-AT under different t_{max}. On the other hand, the attack strength significantly increases along with the increase of ϵ. It can be observed that PrincipaLS consistently demonstrates better robustness under different ϵ. Apparently, PrincipaLS is scalable to different attack budgets.

Stability of latent space. We compute the mean L_2-norm between the latent space of adversarial examples and that of their clean counterpart: $\| \mathbf{Z}_{adv} - \mathbf{Z} \|_2$. As can be seen in Figure 7.6, PrincipaLS's mean L_2-norm is three orders of magnitude smaller than the other defenses. This indicates that PrincipaLS's latent space is barely affected by adversaries, showing PrincipaLS's effectiveness in adversary removal.

Reconstruction errors. For an AE-style novelty detection model, normal data and anomalous data are expected to get low and high reconstruction errors, respectively. The model follows this behavior given clean data, as shown in Figure 7.7 (a). When an attacker attempts to maximize the reconstruction errors of normal data and minimize that of anomalous data, the model would make wrong predictions, shown in Figure 7.7 (b). Figure 7.7 (c) shows that PGD-AT pulls back the enlarged reconstruction errors of normal data, but they still overlap for the anomalous data. In Figure 7.7 (d), it can be

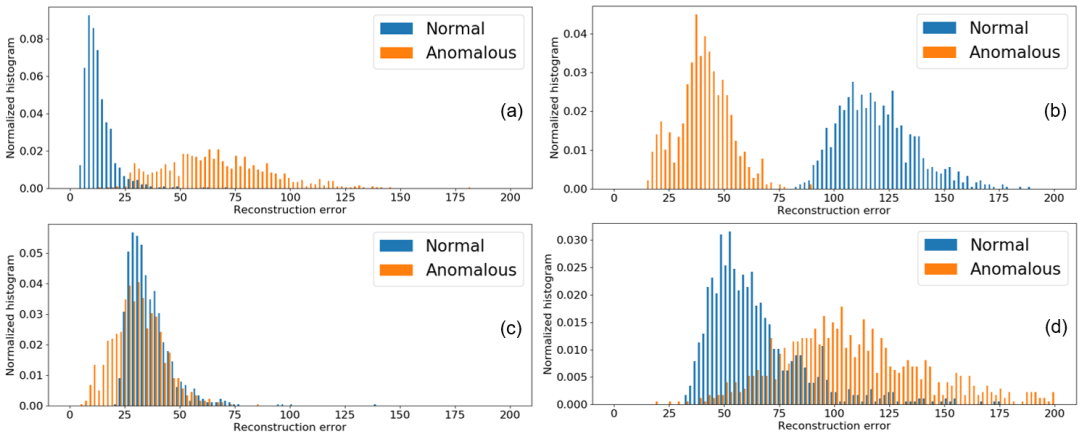

Figure 7.7: Histograms of reconstruction errors. (a) No Defense under clean data. (b) No Defense under PGD attack. (c) PGD-AT under PGD attack. (d) PrincipaLS under PGD attack. Digit 0 of MNIST is set to normal data, and the other digits are anomalous.

observed that PrincipaLS pushes the reconstruction errors of anomalous data with a better margin. Although the reconstruction errors of normal data also increase, the gap between normal and anomalous data is retained resulting in PrincipaLS performing better under attacks.

Reconstructed images. Figure 7.8 compares the reconstructed images of No Defense model and PrincipaLS under PGD and AF attacks. Digit 2 of MNIST is used as the known class. In the PGD case, No Defense model produces decent reconstructions for both adversarial normal and anomalous data. Hence, the reconstruction error gap between normal data and anomalous data is insufficiently large. In the AF case, No Defense model still captures the shape of the adversarial anomalous data and thus produces fair reconstructions, but it fails to reconstruct recognizable patterns for adversarial normal data. Therefore, the resulting reconstruction errors would cause wrong predictions. Such observations are consistent with the quantitative results that it is not adversarially robust. In contrast, PrincipaLS reconstructs every data into the known class of digit 2. Hence, even under attacks, PrincipaLS can obtain very high

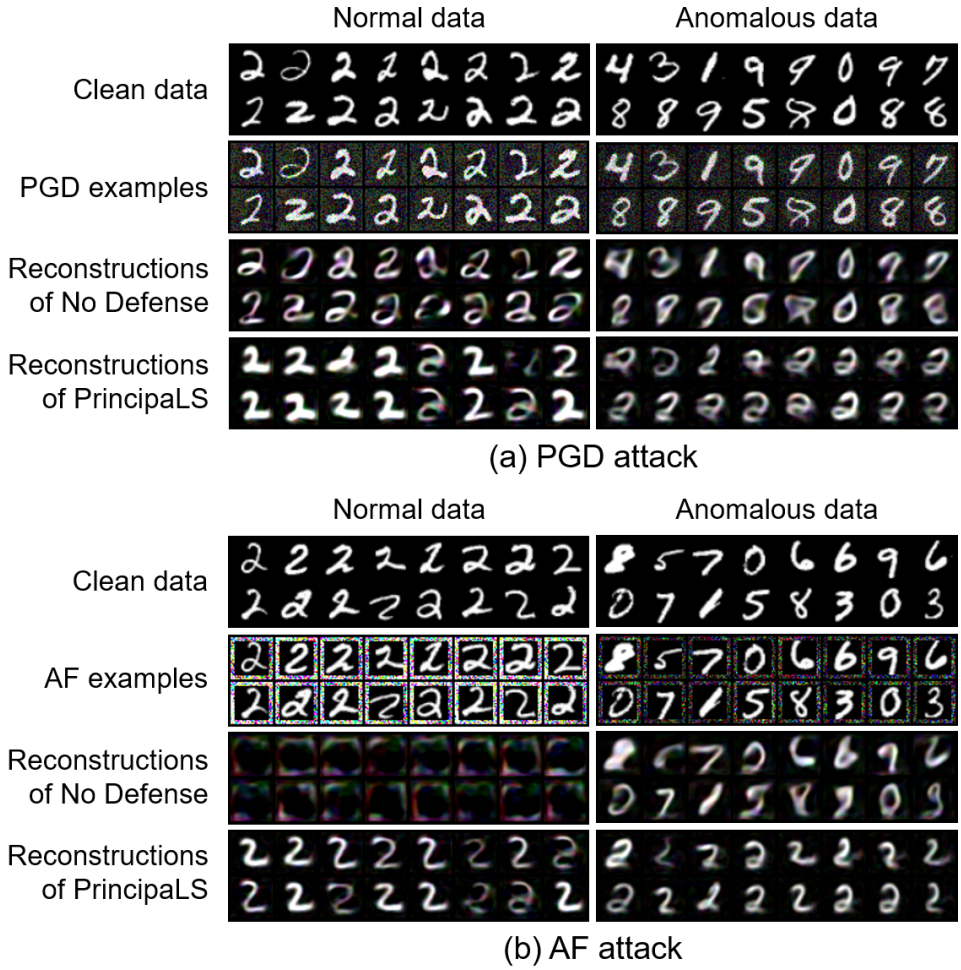

Figure 7.8: Reconstructions under (a) PGD attack with $\epsilon = 76/255$ and (b) AF attack with framing with $= 1$, $\epsilon = 255/255$. Digit 2 is set to normal data, and the other digits are anomalous.

reconstruction errors from anomalous data and low errors from normal data.

Evaluation with FPR at 95% TPR. In addition to the AUROC metric, in Table 7.7, we also provide the mean of FPR at 95% TPR comparison for different defenses on the MNIST dataset [116]. We observe a similar trend as that of mAUROC (see Table 7.1). The proposed PrincipaLS outperforms all the other defense approaches.

Table 7.7: The mean of FPR at 95% TPR under PGD attack.

Defense	Clean	PGD
PGD-AT [150]	0.229	0.912
FD [239]	0.243	0.914
SAT [237]	0.360	0.916
RotNet-AT [72]	0.252	0.909
PrincipaLS (Ours)	**0.170**	**0.803**

7.5 Discussion

7.5.1 Further Comparison with ARAE

ARAE [186] somewhat refers to the adversarial robustness of novelty detection though its main purpose is improving performance. As mentioned in Sec. 7.2, ARAE's adversarial robustness is not thoroughly evaluated. In this section, we make a comprehensive comparison with ARAE.

First, ARAE evaluates adversarial robustness by crafting adversarial examples from only the normal test data (the known class). We refer to this attack as *PGD-normal*. Instead, our attack method crafts adversarial examples from every test data regardless of their class (see Sec. 7.2). We reproduce PGD-normal with the same setting as in Sec. 7.4.1. As shown in Table 7.8, the proposed attack (denoted as PGD) is stronger than PGD-normal, in which PGD obtains lower mAUROC across all the considered defense methods and datasets. It is intuitive that perturbing every input data poses a stronger attack.

Second, ARAE performs AT on the latent space-based adversarial examples. We name this attack as *PGD-latent*. Instead, in this chapter, we perform AT on the output space-based adversarial examples (see Sec. 7.2). We reproduce PGD-latent with the same setting as in Sec. 7.4.1. Specifically, PGD-latent replaces the loss objective

Table 7.8: The mAUROC of models under PGD, PGD-normal, PGD-latent, PGD-clean and PGD-anomalous attacks. Underlines denote the lowest mAUROC, which indicates the strongest attack.

Defense	Attack method	MNIST	F-MNIST	CIFAR-10
No Defense	Clean	0.964	0.892	0.550
	PGD	<u>0.051</u>	<u>0.088</u>	<u>0.034</u>
	PGD-normal	0.167	0.284	0.111
	PGD-latent	0.773	0.715	0.433
	PGD-clean	0.106	0.180	0.070
	PGD-anomalous	0.939	0.788	0.332
PGD-AT [150]	PGD	<u>0.357</u>	<u>0.368</u>	<u>0.145</u>
	PGD-normal	0.745	0.656	0.309
	PGD-latent	0.914	0.784	0.448
	PGD-clean	0.863	0.802	0.403
	PGD-anomalous	0.753	0.677	0.328
FD [239]	PGD	<u>0.366</u>	<u>0.379</u>	<u>0.147</u>
	PGD-normal	0.750	0.654	0.309
	PGD-latent	0.906	0.762	0.447
	PGD-clean	0.871	0.794	0.401
	PGD-anomalous	0.761	0.673	0.331
PrincipaLS (Ours)	PGD	<u>0.706</u>	<u>0.613</u>	<u>0.246</u>
	PGD-normal	0.905	0.786	0.399
	PGD-latent	0.962	0.882	0.547
	PGD-clean	0.936	0.867	0.520
	PGD-anomalous	0.881	0.781	0.407

Eq. (7.2) with follows:

$$\mathcal{L}(\mathbf{X}^t, \mathbf{X}, y) = y \parallel Enc(\mathbf{X}^t) - Enc(\mathbf{X}) \parallel_2, \tag{7.16}$$

where Enc denotes the encoder in an AE. As can be seen in Table 7.8, PGD is much stronger than PGD-latent, in which PGD obtains lower mAUROC across all the considered defense methods and datasets. Therefore, we perform AT by minimizing Eq. (7.2) to make a stronger defense.

Third, a novelty detector would not know whether an input image is adversarial or not during inference. In other words, if the given input is an adversarial image, the clean counterpart is not available at test time. Hence, a novelty detector should

compute the novelty score by the reconstruction error between the reconstructed image and the "input image" (regardless it is clean or adversarial) instead of that between the reconstructed image and the "clean image". For instance, if a given test image is an adversarial example $\mathbf{X}_{adv}$, a novelty detector should compute $\| \hat{\mathbf{X}}_{adv} - \mathbf{X}_{adv} \|_2$ instead of $\| \hat{\mathbf{X}}_{adv} - \mathbf{X} \|_2$ as the novelty score, where $\mathbf{X}$ is the clean image. Therefore, to craft a strong adversarial example, one should maximize the reconstruction error between the reconstructed image and the "input image" (regardless it is clean or adversarial). The proposed attack is based on this nature; that is, at each attack iteration, we maximize the reconstruction error between the current adversarial example and the reconstruction of that current adversarial example (see Eq. (7.2)). We make an attack variant, *PGD-clean*, which maximizes the reconstruction error between the clean image and the reconstruction of the current adversarial example. Specifically, PGD-clean replaces the loss objective Eq. (7.2) with follows:

$$\mathcal{L}(\hat{\mathbf{X}}^t, \mathbf{X}, y) = y \, \| \hat{\mathbf{X}}^t - \mathbf{X} \|_2 \, . \tag{7.17}$$

ARAE uses this form. As shown in Table 7.8, PGD is much stronger than PGD-clean, in which PGD obtains lower mAUROC across all the considered defense methods and datasets. Therefore, we perform AT by minimizing Eq. (7.2) to make a stronger defense.

In summary, the proposed attack is much stronger than PGD-normal, PGD-latent and PGD-clean. Hence, we can carefully and strictly evaluate the adversarial robustness of novelty detectors. Moreover, conducting AT on a stronger attack can enhance robustness to a greater extent, so using the proposed attack for doing AT can make novelty detectors much more robust. We hope to provide researchers with a solid

benchmark for future work on the adversarial robustness of novelty detection.

7.5.2 Further Comparison with APAE

To the best of our knowledge, APAE [61] might be the only present defense designed for anomaly detection. However, as mentioned in Sec. 7.2, APAE's adversarial robustness is not thoroughly evaluated. In this section, we make more comparisons with APAE.

First, APAE evaluates adversarial robustness by crafting adversarial examples from only the anomalous test data (the unknown classes), which is contrary to ARAE's PGD-normal (Sec. 7.5.1). We name this attack *PGD-anomalous*. Instead, our attack method crafts adversarial examples from every test data regardless of their class (see Sec. 7.2). We reproduce PGD-anomalous with the same setting as in Sec. 7.4.1. As shown in Table 7.8, the proposed attack (denoted as PGD) is stronger than PGD-anomalous, in which PGD obtains lower mAUROC across all the considered defense methods and datasets. It is intuitive that perturbing every input data poses a stronger attack. On the other hand, No Defense attains the best mAUROC compared with the other defenses. The reason is that these defenses use only normal data to do AT, so they overfit the adversarial normal data and show less robustness against PGD-anomalous.

Second, APAE claims that AT does not apply to the novelty detection problem. In contrast, in this chapter, we demonstrate that AT actually does apply to novelty detection, in which we can craft adversarial examples for the normal data and use them to train the target model. Indeed, using AT is less robust to PGD-anomalous as shown in Table 7.8. However, for the stronger attacks (i.e., the proposed attack) that contain adversarial normal data, AT can significantly improve the robustness.

Apparently, we construct a more appropriate evaluation protocol to fully test the adversarial robustness of novelty detectors. With a proper evaluation protocol, we are able to design a much better defense method accordingly.

7.5.3 Comparison with the Defenses that Use Dimensionality Reduction Techniques

A few studies employ vanilla PCA to counter adversarial attacks for the image classification problem. Hendrycks & Gimpel [71] and Jere et al. [88] utilized PCA to detect adversarial examples. Li & Li [123] performed PCA in the feature domain and used a cascade classifier to detect adversarial examples. However, detection is inherently weaker than defense in terms of resisting adversarial attacks. Bhagoji et al. [11] mapped each input image into a dimensionality-reduced PCA space to defend against adversarial attacks, but this fails to resist white-box attacks [18]. As discussed in Sec. 7.1, doing image classification requires a model containing sophisticated semantic information, and large manipulation such as dimensionality reduction would hurt the model capability. Hence, it is counterintuitive to use dimensionality reduction for robustifying image classification models.

In contrast, we target a different downstream application, one-class novelty detection. As discussed in Sec. 7.1, novelty detection has a peculiar property in that a novelty detector's latent space can be manipulated to a larger extent as long as it retains the known class information. This is naturally suitable for using dimensionality reduction techniques to remove adversaries and maintain the model capability simultaneously. Furthermore, we propose a novel training scheme that learns the incrementally-trained cascade principal components in the latent space. The proposed defense method is fully differentiable at inference time, and it is highly robust to

white-box attacks as shown in Sec. 7.4.2.

7.5.4 Comparison with Vector Quantization

The proposed PrincipaLS learns a principal latent vector, which is adversary-free, to replace perturbed latent vectors and enhance adversarial robustness. An alternative way of learning the adversary-free latent vectors is using vector quantization. VQ-VAE [215] is an AE variant that uses the vector quantization technique to improve generation ability. To the best of our knowledge, VQ-VAE has not been adopted in the context of novelty detection. In this section, we implement VQ-VAE for one-class novelty detection and evaluate its adversarial robustness. We set the number of embeddings to 4 for MNIST, 8 for F-MNIST and 256 for CIFAR-10. These numbers achieve the best robustness according to our experiments.

Because the quantization step is non-differentiable, it causes obfuscated gradients [5]. Hence, we build a neural network, which consists of four fully connected layers, to learn the mapping from the latent vectors (the output of the encoder) to the quantized latent vectors (corresponding embedding vectors). Since the neural network is differentiable, we use it to approximate the gradients of the non-differentiable part to perform PGD attack [150]. For comparison, we train another neural network with the same architecture to learn the mapping from the latent space to the principal latent space of PrincipaLS.

Table 7.9 reports the experimental results. Comparing PrincipaLS (PGD examples are generated from the entire differentiable network) and PrincipaLS* (PGD examples are generated from the neural network gradient approximator), we can see that the neural network still cannot perfectly approximate the gradients, so the produced attack

Table 7.9: The mAUROC of VQ-VAE and PrincipaLS under PGD attack. "*" denotes that PGD examples are generated from a neural network gradient approximator.

Defense	MNIST	F-MNIST	CIFAR-10
VQ-VAE [215]*	0.542	0.588	0.248
PrincipaLS (Ours)	0.706	0.613	0.246
PrincipaLS (Ours)*	0.816	0.755	0.325

is weaker. However, although attacked by this weaker attack, VQ-VAE achieves lower mAUROC than PrincipaLS on MNIST and F-MNIST, and much lower mAUROC than PrincipaLS* on all the datasets. This shows that PrincipaLS has better robustness than VQ-VAE.

The explanations are as follows. First, PrincipaLS's principal latent vector is learned by the incrementally-trained cascade PCA process, which is not only adversary-free but also contains important features that can properly substitute the original latent vectors. In contrast, VQ-VAE's embedding vectors are randomly initialized. Even using the training strategy in [215], the embedding vectors are still not close to the original latent vectors. Therefore, PrincipaLS's principal latent vector is a better adversary-free substitute. Second, after Vector-PCA, PrincipaLS's Vector-PCA map stores the scaling factors of the principal latent vector with spatial information, so we can perform Spatial-PCA on it to further remove the remaining adversaries. In contrast, the vector quantization map stores the indices of the embedding vectors, and we cannot do further operations on these indices. These demonstrate the advantages of the proposed PrincipaLS.

Table 7.10: The AUROC of multi-class novelty detection on MNIST. Digit 0 and digit 2 are set to the known classes.

Defense	Clean	PGD
No Defense	0.926	0.051
PGD-AT [150]	0.926	0.052
FD [239]	0.926	0.066
PrincipaLS (Ours)	**0.954**	**0.412**

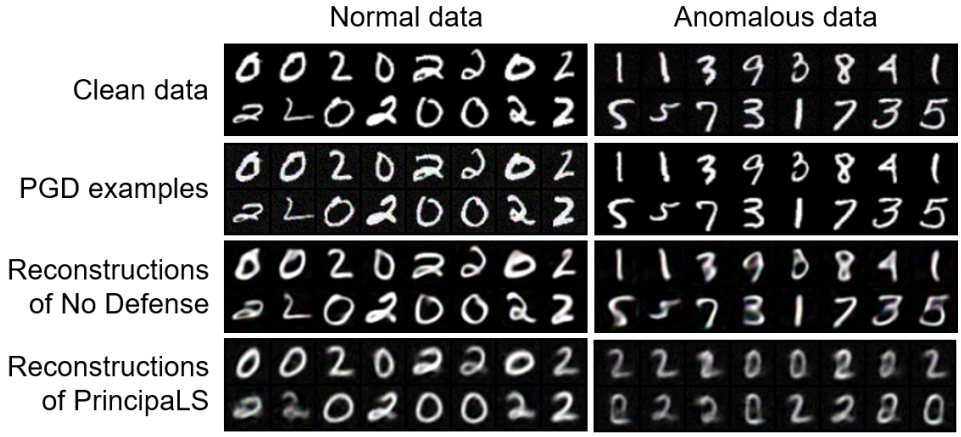

Figure 7.9: Reconstructions under PGD attack with $\epsilon = 25/255$. Digit 0 and digit 2 are set to normal data, and the other digits are anomalous.

7.5.5 Applying to Multi-Class Novelty Detection

Multi-class novelty detection [162, 169] has the same problem setting as one-class novelty detection except that it considers multiple known classes. It is more challenging, as it needs to characterize the underlying distributions of multiple known classes and identify novel classes given such knowledge of multiple known classes. We explore applying the proposed PrincipaLS to multi-class novelty detection.

In this experiment, we define digit 0 and digit 2 of MNIST dataset as the known classes (normal data), and the rest of the digits are novel classes (anomalous data). In Table 7.10, we can find that PGD-AT and FD do not improve adversarial robustness,

which demonstrates that image classification-based defense approaches cannot protect multi-class novelty detectors. In contrast, PrincipaLS significantly increases PGD accuracy, showing its potential for applying to multi-class novelty detection. Figure 7.9 shows that PrincipaLS reconstructs every normal and anomalous data into the known classes of digit 0 or digit 2. Therefore, even under adversarial attacks, PrincipaLS can obtain very high reconstruction errors from anomalous data and low errors from normal data. Furthermore, PrincipaLS achieves higher clean accuracy, as the principal latent components can exclusively characterize the latent space of the known classes (see Sec. 7.4.5).

7.5.6 Applying to Image Classification

The proposed PrincipaLS method is specifically designed for the novelty detection task. As discussed in Sec. 7.1 and Sec. 7.3, it leverages the task-specific knowledge that novelty detectors are only required to retain information about normal data, thereby resulting in the preferred high reconstruction errors for anomalous data. This property allows PrincipaLS to largely manipulate the latent space of novelty detectors to remove adversaries, while maintaining the performance on clean data. It can be noted that such property is unique to novelty detection, as most visual recognition problems (e.g., image classification) require a model containing high-level semantic information. Hence, a large manipulation on the latent space would limit the model capacity and thus degrade accuracy.

To demonstrate this, we apply PrincipaLS to the image classification task on CIFAR-10 [103]. We attach the PrincipaLS module to ResNet-18 [69] between the last convolutional layer and the fully-connected layer. We use PGD attack with

Table 7.11: Image classification accuracy (%) on CIFAR-10.

Defense	Clean	PGD
No Defense	94.0	0.0
PGD-AT [150]	83.3	51.2
FD [239]	83.3	51.5
PrincipaLS $k_V = 1$	36.0	30.6
PrincipaLS $k_V = 8$	71.1	47.0
PrincipaLS $k_V = 64$	72.7	48.3

$\epsilon = 8/255$ for testing and AT. Table 7.11 shows that PGD-AT and FD, which are originally designed for image classification, effectively improve adversarial robustness. As expected, PrincipaLS obtains both lower clean and PGD accuracies. The reason is that the PrincipaLS operation reduces the model capacity for learning high-level semantic representations, making the latent space insufficiently discriminative for classification. We can see that larger k_V achieves higher PGD accuracy, which is an opposite trend to that in novelty detection (see Table 7.6). In other words, different from novelty detectors, image classifiers cannot enjoy the principal latent space since it loses too much semantic information.

PrincipaLS, designed for novelty detection, does not work on image classification; conversely, the defenses designed for image classification are not that effective on novelty detection as shown in Table 7.1 and Table 7.2. Apparently, these two vision tasks have different characteristics and thus need different adversarial defenses. This demonstrates the need for a defense method specifically designed for novelty detection and thus highlights the contribution of this chapter.

7.5.7 Sanity Checks to Evaluation

To further verify that the proposed PrincipaLS's robustness is not due to obfuscated gradients, we report our results on the basic sanity checks introduced in [5]:

- Table 7.1 shows that iterative attacks (PGD and MI-FGSM) are stronger than one-step attacks (FGSM).
- Table 7.1 shows that white-box attacks are stronger than black-box attacks (by MI-FGSM).
- Unbounded attacks reach 100% attack success rate (AUROC drops to 0.000) on all the five datasets.
- Figure 7.5 shows that increasing distortion bound increases attack success (decreases AUROC).

7.6 Summary

In this chapter, we study the adversarial robustness in the context of the one-class novelty detection problem. We show that existing novelty detection models are vulnerable to adversarial perturbations and then propose a defense method referred to as PrincipaLS. Specifically, PrincipaLS purifies the latent space by the incrementally-trained cascade PCA process. Moreover, we construct a generic evaluation framework to fully test the effectiveness of the proposed PrincipaLS. We perform extensive experiments on multiple datasets with multiple existing novelty detection models and consider various attacks to show that PrincipaLS consistently improves adversarial robustness.

Part II

Robust Computer Vision Against Domain Shifts

Chapter 8

Learning Feature Decomposition for Domain Adaptive Monocular Depth Estimation

8.1 Motivation

Depth information is essential to many robotic applications, e.g., localization, mapping and obstacle detection. Existing depth acquisition devices, such as Lidar and structured-light sensors, are typically bulky, heavy and power-consuming. Therefore, they are unsuitable for compact robotic platforms. This motivates the progress of Monocular Depth Estimation (MDE) that predicts depth from a single image, as it has low cost, small size, high power efficiency, and no need to re-calibrate after a long time of use.

Recent advances in deep learning have enabled supervised learning approaches to perform MDE [12, 44, 48, 115], but obtaining ground-truth depth annotations is costly and labor-intensive. Moreover, if we can only obtain the depth annotations corresponding to a specific camera for training, the domain shift problem would happen when we test the trained model on another camera's data (see details in Chapter 2). These challenges hinder the MDE technique from applying to compact robotic platforms. Hence, developing algorithms that can transfer the knowledge learned from one labeled dataset to another unlabeled dataset becomes increasingly important.

We approach this via UDA. Existing works mainly rely on a synthetic-to-real translation or vice versa to bridge the domain gap [2, 4, 29, 148, 171, 259, 260]. Although these works have achieved great improvements, image translation itself is not an easy task. Images may not be perfectly translated to another domain or contain distortion after translation. Another research stream performs feature alignment through adversarial learning [29, 109, 171, 260]. Nevertheless, it is difficult to completely align the entire feature space from different domains owing to the domain

shift problem.

To overcome these challenges, inspired by recent approaches [171, 260] and disentangled learning techniques [21, 112, 117, 119, 130], we assume that the feature space can be decomposed into content and style components. The content component consists of semantic features that are shared across different domains. For example, consider images of indoor scenes from two different datasets. Objects like tables, chairs and beds are content information. Such semantic features are more domain-invariant, so it is easier to align the content component from different domains. In contrast, the style component is domain-specific. For instance, style features like texture and color are unique to the scenes captured by a particular camera, so aligning the style features may not be practical. Hence, to train a MDE model working for the target data, we suggest discarding the source-specific style component that hinders adaptation to narrow the domain gap, but including the target-specific style component that is still useful for the primary MDE task.

Based on the above intuitions, we propose a novel UDA method for the MDE task, referred to as Learning Feature Decomposition for Adaptation (LFDA): (1) Different from prior works attempting to align the entire feature maps of source and target data [29, 109, 171, 260], LFDA only needs to align the content features that already have a much smaller domain gap. (2) To further improve the content feature alignment, LFDA individually estimates the statistics of different feature domains via separate BN [22, 139, 236], which can bypass the domain-specific elements in the feature space. The separate BN structure also helps to properly integrate the content and style features of the target data. (3) With the proposed decomposition learning, LFDA bridges the domain gap more efficiently. In particular, it keeps a relatively compact

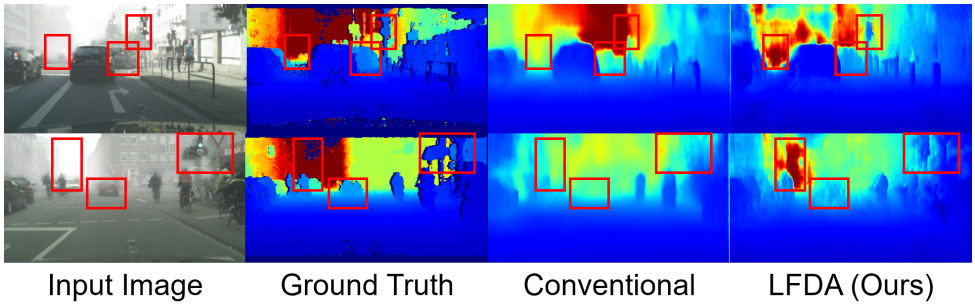

Figure 8.1: Example results of domain adaptive MDE on the Foggy Cityscapes dataset [184]. It is a scenario of adverse weather adaptation. "Conventional" refers to the method based on the usual domain adversarial learning [52]. The red boxes highlight regions where our method makes improvements.

structure at inference time, leading to lower computational complexity compared to the recent advances which require a sophisticated image translation network during inference [171, 259]. (4) In addition, most existing approaches rely on a multi-stage training procedure that first pre-trains each sub-networks separately then fine-tunes them together [2, 4, 148, 171, 259]. Instead, LFDA is trained end-to-end in a single stage, making it more feasible to deploy in practical applications.

In evaluation, the majority of existing studies only focus on synthetic-to-real adaptation [2, 4, 29, 109, 171, 259, 260]. In contrast, we apply our method to three broad scenarios of domain adaptation: (1) cross-camera adaptation, (2) synthetic-to-real adaptation, and (3) adverse weather adaptation [220]. To the best of our knowledge, this chapter is the first attempt that considers all the three scenarios for the MDE task. Particularly, adverse weather adaptation is the first time explored for MDE. Figure 8.1 shows examples of adverse weather adaptation results. Compared to a conventional approach, our LFDA can obtain more accurate depth predictions for cars, traffic signs, sky, *etc.*, under foggy weather conditions. More extensive experiments in Sec. 8.3 demonstrate that LFDA achieves promising performance in all the scenarios.

8.1.1 Related Work

Deep learning has achieved high accuracy for MDE by supervised learning. Eigen et al. [44] introduced a deep learning-based MDE approach with a multi-scale network. Afterward, Laina et al. [115] presented a deeper network with a fully convolutional network and residual learning. Fu et al. [48] divided depth ranges into multiple depth bins and solved MDE in a classification manner using an ordinal regression loss. Recently, Bhat et al. [12] developed a transformer-based block to adaptively adjust the depth bins for each image. Several studies explore training MDE models via self-supervision. Notable algorithms include exploiting epipolar geometry constraints from stereo pairs [54, 57, 111] and utilizing multi-view information from monocular video sequences [152, 262].

Domain adaptation for MDE is first introduced by Atapour et al. [4], where they train a depth estimation network using synthetic images then translated real images to synthetic style during inference. AdaDepth [109] employs adversarial learning at both feature and output spaces to align the distributions between the source and target domains. T^2Net [260] transfers synthetic images to real style to train a MDE network. CrDoCo [29] and GASDA [259] use bidirectional style transfer to learn the mapping between two domains, where GASDA also exploits epipolar geometry structure for real images. SharinGAN [171] translates both synthetic and real data to a single shared domain to decrease their discrepancy. DESC [148] leverages an additional semantic segmentation network and edge detection to provide semantic and edge guidance. Akada et al. [2] adopt recent self-supervised learning techniques to learn domain-invariant representations. However, they either suffer from sub-optimal domain alignment or high computational complexity during inference.

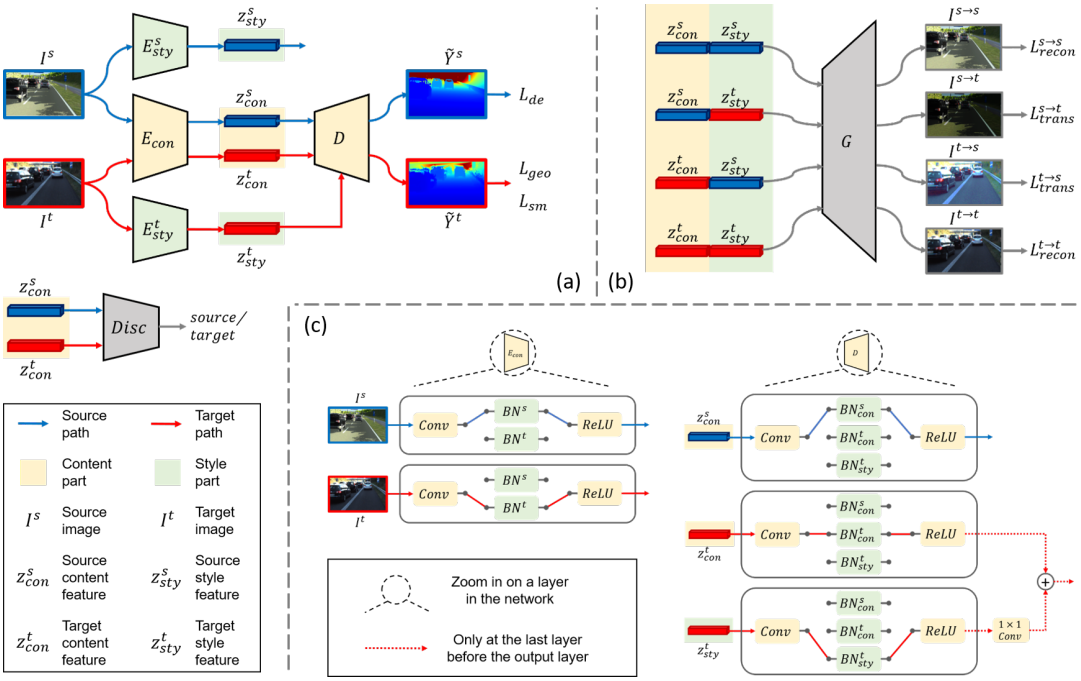

Figure 8.2: Overview of the proposed LFDA framework. E_{con}: shared content encoder, E^s_{sty}: source-specific style encoder, E^t_{sty}: target-specific style encoder, D: depth estimation task decoder, G: generator, and $Disc$: domain discriminator. (a) Main information flow. (b) Learning translations for feature decomposition. (c) Separate BN structure for feature alignment and integration.

8.2 Proposed Method

8.2.1 Framework

An overview of the proposed LFDA is shown in Figure 8.2. The entire framework consists of eight sub-networks: shared content encoder E_{con}, source-specific style encoder E^s_{sty}, target-specific style encoder E^t_{sty}, MDE task decoder D, generator G, domain discriminator $Disc$, source-to-target translation discriminator $Disc^{s\to t}$, and target-to-source translation discriminator $Disc^{t\to s}$. $\{E_{con}, D\}$ composes as a MDE primary task network, which is a standard encoder-decoder architecture.

Feature decomposition. As illustrated in Figure 8.2 (a), the two individual style

encoders E^s_{sty} and E^t_{sty} extract the domain-specific style features of the given source input I^s and target input I^t, respectively. This is formulated as $z^s_{sty} = E^s_{sty}(I^s)$ and $z^t_{sty} = E^t_{sty}(I^t)$. We believe that the content of images is more domain-invariant, so a shared content encoder E_{con} is used to learn the content features of both source and target images, formulated as $z^s_{con} = E_{con}(I^s)$ and $z^t_{con} = E_{con}(I^t)$. This decomposition is achieved by the training scheme shown in Figure 8.2 (b), and the details are elaborated in Sec. 8.2.2.

Feature alignment. Although the content features z^s_{con} and z^t_{con} learned by a standard encoder already have a small domain gap, they are still not completely domain-invariant, as the content of images from different domains also contains some domain-specific elements, such as scale and viewpoint. To address this, we perform feature alignment in two aspects.

First, we propose to estimate the feature distributions of I^s and I^t individually using a separate BN structure [22, 139, 236]. Specifically, two BN branches [82], denoted as BN^s and BN^t, are deployed after each convolutional layer in E_{con} (see Figure 8.2 (c)). Each BN branch works individually for its own domain. To elaborate, BN^s and BN^t learn domain-specific affine parameters $\{\gamma^s, \beta^s\}/\{\gamma^t, \beta^t\}$, and distribution statistics $\{\mu^s, \sigma^s\}/\{\mu^t, \sigma^t\}$ for the source and target data, respectively. Note that all the layers other than BNs are still shared (e.g., convolution and ReLU). Suppose that $\ddot{z}^d_{con}$ is the content feature of domain d, where $d \in \{s, t\}$, the separate BN structure at an arbitrary layer in E_{con} is formulated as:

$$BN^d(\ddot{z}^d_{con};\, \gamma^d, \beta^d) = \gamma^d \left(\frac{\ddot{z}^d_{con} - \mu^d}{\sqrt{(\sigma^d)^2 + k}} \right) + \beta^d, \tag{8.1}$$

where k is a tiny constant for numerical stability. With this design, the domain gap

between z^s_{con} and z^t_{con} is acquired by the domain-specific parameters $\{\mu^d, \sigma^d, \gamma^d, \beta^d\}$, and their domain-invariant part passes through each BN layer.

Second, inspired by GRL [52], we employ adversarial learning [59] to align the features z^s_{con} and z^t_{con} (see Figure 8.2 (a)). Details are discussed in Sec. 8.2.2.

Feature integration. Our feature decomposition extracts four preferred components: $\{z^s_{con}, z^s_{sty}, z^t_{con}, z^t_{sty}\}$, where z^s_{con} and z^t_{con} are aligned by our separate BN structure and adversarial learning. To train the MDE task decoder D, we use z^s_{con}, z^t_{con} and z^t_{sty}. We discard z^s_{sty} since it is specific to source data and thus cannot help the model adapt to the target domain. Instead, the target-specific style component z^t_{sty} is still useful for the MDE model that works for the target domain.

After feature decomposition, z^t_{sty} and z^t_{con} have different feature characteristics though they are from the same target domain. Hence, directly fusing them in the task decoder D would cause potential accuracy degradation. To address this issue, as shown in Figure 8.2 (c), we also deploy separate BNs in D. There are three BN branches: BN^s_{con}, BN^t_{con} and BN^t_{sty}, each of which works as Eq. (8.1). BN^s_{con} and BN^t_{con} are used for the same purpose as discussed before, and BN^t_{sty} is responsible for characterizing the feature distribution of z^t_{sty} exclusively. Since the content and style features have different underlying distributions, simply leveraging a single set of BN parameters for z^t_{con} and z^t_{sty} would estimate an inaccurate mixture. Therefore, the additional BN^t_{sty} is used to disentangle such mixture distribution, allowing proper integration of z^t_{con} and z^t_{sty} for decoding target features. Because the content and style components may have different importance for MDE, we employ a 1×1 convolution and a residual connection to combine z^t_{con} and z^t_{sty} right before the output layer of D. This weighted fusion helps to adjust the balance between these two features of target

data. Finally, D outputs predicted depth maps, $\tilde{Y}^s = D(z^s_{con})$ and $\tilde{Y}^t = D(z^t_{con}, z^t_{sty})$, respectively.

8.2.2 Objectives

The proposed LFDA framework is trained with the following objective functions.

Feature decomposition loss. This loss is used to decompose the feature components according to our assumption for domain adaptation. It consists of translation loss and reconstruction loss.

Inspired by style transfer techniques [79, 95], we adopt the translation loss to separate the content and style features of an input image. Let us consider the case of source-to-target image translation in our framework. Given a source image I^s and a target image I^t, we aim to derive a translated image $I^{s\to t} = G(z^s_{con}, z^t_{sty})$ which consists of the content of I^s and the style of I^t (see Figure 8.2 (b)). We achieve this translation via objective $\mathcal{L}^{s\to t}_{trans}$, which consists of two perceptual losses [95] and an adversarial loss:

$$\begin{aligned}\mathcal{L}^{s\to t}_{trans} = &\sum_{j\in L} w^{trans}_{con,j} \left\| \phi_j(I^s) - \phi_j(I^{s\to t}) \right\|_1 \\ &+ \sum_{j\in L} w^{trans}_{sty,j} \left\| \mu(\phi_j(I^t)) - \mu(\phi_j(I^{s\to t})) \right\|_1 \\ &+ \eta \left(Disc^{s\to t}(I^{s\to t}) - 1 \right)^2,\end{aligned} \tag{8.2}$$

where $\eta = 0.2$, w^{trans}_{con} and w^{trans}_{sty} are pre-defined weights, L denotes the $\{relu1\_1, relu2\_1, relu3\_1, relu4\_1, relu5\_1\}$ layers of a pre-trained VGG network [195] that measures perceptual loss, ϕ_j is the j-th layer in L, and $\mu(\cdot)$ returns the channel-wise mean values of a feature space. This translation loss has also been

explored by [21].

To elaborate, the first perceptual loss computes the distance of the high-level content features between I^s and $I^{s\to t}$ such that $I^{s\to t}$ contains the content of I^s. Since the content information mostly exists in higher layers of VGG, we set w_{con}^{trans} to $\{0, 0, 0, 1/4, 1\}$. The second perceptual loss forces $I^{s\to t}$ to contain the style of I^t. To explicitly encode the style information of an image, we employ AdaIN structure [79] that measures the distance of the channel-wise mean values of the style features between I^t and $I^{s\to t}$. Since the style information mostly exists in lower layers of VGG, we set w_{sty}^{trans} to $\{1, 1, 1, 0, 0\}$. The third term is a standard least-squares adversarial loss [155], where we assign labels 1 and 0 to untranslated and translated images, respectively. This loss helps to improve the quality of image translation. As for the case of target-to-source translation, it is symmetric to source-to-target translation. We define its objective as $\mathcal{L}_{trans}^{t\to s}$, which replaces s to t, t to s and $s \to t$ to $t \to s$ in Eq. (8.2).

The reconstruction loss is used to guarantee that the combination of the decomposed content and style components forms a nearly complete representation of an input image [21]. Let us consider the case of source image reconstruction. Given a source image I^s, we aim to derive a reconstruction $I^{s\to s} = G(z_{con}^s, z_{sty}^s)$ (see Figure 8.2 (b)). This can be achieved via objective $\mathcal{L}_{recon}^{s\to s}$, which is also based on the perceptual loss:

$$\mathcal{L}_{recon}^{s\to s} = \sum_{j\in L} w_j^{recon} \left\| \phi_j(I^s) - \phi_j(I^{s\to s}) \right\|_1, \tag{8.3}$$

where $w^{recon} = \{1/32, 1/16, 1/8, 1/4, 1\}$. Symmetrically, target image reconstruction is achieved via objective $\mathcal{L}_{recon}^{t\to t}$, which replaces s to t and $s \to s$ to $t \to t$, from Eq. (8.3).

With the above loss functions, LFDA decomposes the feature space into

$\{z^s_{con}, z^s_{sty}, z^t_{con}, z^t_{sty}\}$, where each of which contains its supposed information exclusively.

Feature alignment loss. Different from prior works that attempt to align the entire features [29, 109, 171, 260], LFDA only needs to align the content features that already have a much smaller domain gap, which is easier to achieve. Inspired by GRL [52], we use a domain adversarial loss $\mathcal{L}_{align}$ to align the distributions of z^s_{con} and z^t_{con} (see Figure 8.2 (a)). This is defined as: $\mathcal{L}_{align} = \big(Disc(z^s_{con})\big)^2 + \big(Disc(z^t_{con}) - 1\big)^2$, where we assign labels 1 and 0 to the source and target domain, respectively. We use the least-squares adversarial loss [155] because it is shown to be more stable at training time. Eventually, $\mathcal{L}_{align}$ further reduces the discrepancy between z^s_{con} and z^t_{con}.

Depth estimation loss. This is the primary task objective for MDE. We employ L_1 loss to make use of the source data annotations: $\mathcal{L}^s_{de} = \|\tilde{Y}^s - Y^s\|_1$, where $\tilde{Y}^s = D(z^s_{con})$ is the predicted depth map and Y^s is the corresponding ground-truth. Following GASDA [259] and SharinGAN [171], depth smoothness loss $\mathcal{L}_{sm}$ and geometry consistency loss $\mathcal{L}_{geo}$ are used as self-supervisions for the target data. They are defined as: $\mathcal{L}_{sm} = e^{-\nabla I^t}\|\nabla \tilde{Y}^t\|_1$, where $\tilde{Y}^t = D(z^t_{con}, z^t_{sty})$ is the predicted depth map; $\mathcal{L}_{geo} = \alpha\big(1 - SSIM(I^t, \hat{I}^t)\big) + \beta\|I^t - \hat{I}^t\|_1$, where $\alpha = 0.425$, $\beta = 0.15$, $\hat{I}^t$ is the inverse warped image derived from $\tilde{Y}^t$ the right counterpart of I^t, and SSIM [226] is an image quality metric. Moreover, inspired by image translation-based adaptation approaches [2, 29, 148, 259, 260], we leverage $I^{s\to t}$ that is generated during feature decomposition learning, to adapt the task network to the target domain (i.e., feed $I^{s\to t}$ produced from Figure 8.2 (b) into the pipeline of Figure 8.2 (a)). This is defined as: $z^{s\to t}_{con} = E_{con}(I^{s\to t})$, $z^{s\to t}_{sty} = E^t_{sty}(I^{s\to t})$, and $\hat{Y}^{s\to t} = D(z^{s\to t}_{con}, z^{s\to t}_{sty})$. Then, the L_1 loss is

used to train with the translated image: $\mathcal{L}_{de}^{s\to t} = \|\tilde{Y}^{s\to t} - Y^s\|_1$.

Full learning objective. The full objective of the proposed LFDA framework is defined as:

$$\begin{aligned}\mathcal{L} = (\mathcal{L}_{de}^{s} + \mathcal{L}_{de}^{s\to t}) + \lambda_{geo}\mathcal{L}_{geo} + \lambda_{sm}\mathcal{L}_{sm} + \lambda_{align}\mathcal{L}_{align} \\ + \lambda_{recon}(\mathcal{L}_{recon}^{s\to s} + \mathcal{L}_{recon}^{t\to t}) + \lambda_{trans}(\mathcal{L}_{trans}^{s\to t} + \mathcal{L}_{trans}^{t\to s}),\end{aligned} \tag{8.4}$$

where λ's are trade-off factors. We optimize this loss function end-to-end in a single stage.

8.2.3 Inference

During inference, our goal is to predict a depth map from a given target image. This corresponds to the red path in Figure 8.2 (a). Therefore, only E_{con}, E_{sty}^{t} and D are retained after training, where E_{sty}^{t} is the only required sub-network in addition to the MDE primary task network $\{E_{con}, D\}$. Compared to recent top-performing approaches which require an entire sophisticated image translation network during inference [171, 259], LFDA allows much lower computational complexity. This is attributed to the proposed decomposition learning that reduces the domain gap more efficiently.

8.3 Experiments

We extensively evaluate the proposed LFDA on three domain adaptation scenarios: cross-camera adaptation, synthetic-to-real adaptation, and adverse weather adaptation [220]. Moreover, we conduct an ablation study and analyze the computational complexity of the models.

Table 8.1: Results of Cityscapes-to-KITTI adaptation, tested on KITTI Eigen split (cap 80m). The 1.25^n columns refer to the standard $\delta < 1.25^n$ accuracy metrics.

Method	Lower, better				Higher, better		
	abs-rel	sq-rel	rmse	rmse-log	1.25	1.25^2	1.25^3
T^{2}Net [260]	0.173	1.335	5.640	0.242	0.773	0.930	0.970
DESC [148]	0.149	0.967	5.236	0.223	0.810	0.940	0.976
LFDA (Ours)	**0.119**	**0.963**	**5.049**	**0.207**	**0.855**	**0.948**	**0.977**

8.3.1 Implementation Details

For fair comparison, the architectures of sub-networks $\{E_{con}, D\}$, E^s_{sty}, E^t_{sty}, $Disc$, $Disc^{t\to s}$ and $Disc^{s\to t}$ are implemented identical to the corresponding ones in T^{2}Net [260]. Besides, generator G is implemented as in [21]. The models are trained by Adam optimizer [101] with initial learning rates of $1e^{-4}$ for $\{E_{con}, D\}$ and $2e^{-5}$ for the other sub-networks. The learning rates decrease according to the polynomial decay policy. We set $\lambda_{geo} = 1$, $\lambda_{sm} = \lambda_{align} = 0.01$, $\lambda_{recon} = 0.5$, and $\lambda_{trans} = 0.05$. The entire framework is trained end-to-end in a single stage. The experiments are implemented by PyTorch [166] and conducted on a single NVIDIA Tesla V100 GPU. We will release our source code after the chapter gets accepted.

8.3.2 Cross-Camera Adaptation

Different cameras may have distinct intrinsic parameters or viewpoints, making the captured images have different scales, fields of view, etc. Such domain gap could cause sub-optimal adaptation performance.

Datasets. We use Cityscapes [32] as the source dataset and KITTI [55] as the target dataset. The KITTI Eigen split [44] is used for testing. Following [260], we rescale the input size of KITTI images from 375×1242 to 192×640, and upsample the predicted

depth maps to the original size for evaluation. For Cityscapes, we follow [148] that crops and resizes the images from 1024×2048 to 192×640. The ground-truth depth is capped at 80m.

Results. Table 8.1 reports the results adhered to a standard evaluation protocol [44]. The impressive improvements on all the metrics show the superiority of our LFDA. In particular, LFDA's abs-rel error is 20% lower than DESC [148]. This indicates that the proposed learning of feature decomposition is effective to reduce the domain gap between the images captured by different cameras.

8.3.3 Synthetic-to-Real Adaptation

The style and appearance of synthetic images are usually different from that of real images. This can negatively impact the accuracy on real data.

Datasets. We use Virtual KITTI (vKITTI) [51] and KITTI as the source and the target domains, respectively. Following [260], we resize the vKITTI images to 192×640 and cap the ground-truth depth at 80m. We evaluate on both KITTI Eigen split and KITTI stereo 2015 dataset [157].

Results. Table 8.2 reports the test results on KITTI stereo 2015 dataset. We also put our Cityscapes-to-KITTI model for comparison. As it can be observed, both our models achieve much better accuracy than present approaches in most metrics. Note that Atapour et al. [4] uses the images captures from the GTA5 game as their source data, and KITTI has a smaller domain shift with GTA5 than Cityscapes or vKITTI. Table 8.3 shows the test results on KITTI Eigen split. LFDA significantly outperforms most existing works, while it is behind SharinGAN [171] by a slim margin. Note that SharinGAN requires a sophisticated image translation network

Table 8.2: Results of X-to-KITTI adaptation, tested on KITTI stereo 2015. Top-2 methods are in bold. vK: Virtual KITTI, K: KITTI, CS: Cityscapes, G: GTA5 images.

Method	Dataset	Lower, better				Higher, better		
		abs-rel	sq-rel	rmse	rmse-log	1.25	1.25^2	1.25^3
Atapour et al. [4]	G → K	0.101	1.048	5.308	0.184	0.903	**0.988**	**0.992**
GASDA [259]	vK → K	0.106	**0.987**	5.215	0.176	0.885	0.963	**0.986**
LFDA (Ours)	CS → K	**0.092**	1.055	**5.024**	**0.165**	**0.906**	0.966	0.985
LFDA (Ours)	vK → K	**0.087**	**0.931**	**4.765**	**0.162**	**0.910**	**0.968**	**0.986**

Table 8.3: Results of vKITTI-to-KITTI adaptation, tested on KITTI Eigen split (cap 80m). Top-2 methods are in bold.

Method	Lower, better				Higher, better		
	abs-rel	sq-rel	rmse	rmse-log	1.25	1.25^2	1.25^3
AdaDepth [109]	0.214	1.932	7.157	0.295	0.665	0.882	0.950
CrDoCo [29]	0.232	2.204	6.733	0.291	0.739	0.883	0.942
T^2Net [260]	0.173	1.396	6.041	0.251	0.757	0.916	0.966
Akada et al. [2]	0.168	1.228	5.498	0.235	0.771	0.921	0.973
DESC [148]	0.156	1.067	5.628	0.237	0.787	0.924	0.970
GASDA [259]	0.149	1.003	4.995	0.227	0.824	0.941	0.973
SharinGAN [171]	**0.116**	**0.939**	**5.068**	**0.203**	**0.850**	**0.948**	**0.978**
LFDA (Ours)	**0.120**	**0.961**	**5.095**	**0.213**	**0.848**	**0.945**	**0.975**

during inference, resulting in a much higher computational cost than Ours. Also, it relies on a complicated multi-stage training procedure. Both drawbacks make it unfriendly to be deployed in real-world applications.

8.3.4 Adverse Weather Adaptation

Adverse weather such as fog and rain produce image artifacts. These artifacts can result in accuracy degradation.

Datasets. In this experiment, Foggy Cityscapes [184] is used as the target dataset. It is constructed by simulating haze upon Cityscapes images. We crop and resize the images to 192×640, and cap the ground-truth depth at 80m.

Table 8.4: Results on Foggy Cityscapes (cap 80m).

Method	Dataset	Lower, better				Higher, better		
		abs-rel	sq-rel	rmse	rmse-log	1.25	1.25^2	1.25^3
Src-Only	CS	0.477	8.333	18.211	0.717	0.225	0.507	0.720
Src+Tgt+AL	CS & K	0.422	4.672	11.879	0.448	0.249	0.698	**0.915**
LFDA (Ours)	CS & K	**0.283**	**3.485**	**11.261**	**0.381**	**0.479**	**0.835**	0.914
Src-Only	vK	0.415	9.117	17.356	0.673	0.370	0.631	0.741
Src+Tgt+AL	vK & K	0.378	6.130	15.434	0.600	0.325	0.688	0.795
LFDA (Ours)	vK & K	**0.332**	**4.454**	**13.024**	**0.475**	**0.374**	**0.762**	**0.868**

Table 8.5: Results of ablation study, tested vKITTI-to-KITTI adaptation on KITTI Eigen split (cap 80m).

Method	Lower, better				Higher, better		
	abs-rel	sq-rel	rmse	rmse-log	1.25	1.25^2	1.25^3
Src-Only	0.212	2.196	7.114	0.323	0.673	0.851	0.930
+Tgt+AL	0.140	1.022	5.131	0.216	0.834	0.943	**0.977**
+Tgt+Con+2BN	0.123	1.039	5.220	0.215	0.847	0.944	0.974
+Tgt+Con+2BN+Sty	0.273	3.566	8.371	0.314	0.659	0.882	0.948
LFDA (Ours)	**0.120**	**0.961**	**5.095**	**0.213**	**0.848**	**0.945**	0.975

Results. Table 8.4 reports the test results on Foggy Cityscapes. We evaluate both our models of Cityscapes-to-KITTI and vKITTI-to-KITTI. Since this is the first time in the literature to explore adverse weather adaptation for MDE, we build our own baselines to compared with. Src-Only refers to the model trained on only the source data, and Src+Tgt+AL is trained on both source and target data by adversarial learning to align their entire feature distributions. Clearly, LFDA makes considerable improvements over both baselines, indicating that it performs more stably under different weather conditions. Examples of qualitative results are shown in Figure 8.1.

8.3.5 Ablation Study

We conduct an ablation study using our model of vKITTI-to-KITTI and evaluate on the KITTI Eigen split. The results are reported in Table 8.5. First, we can see that

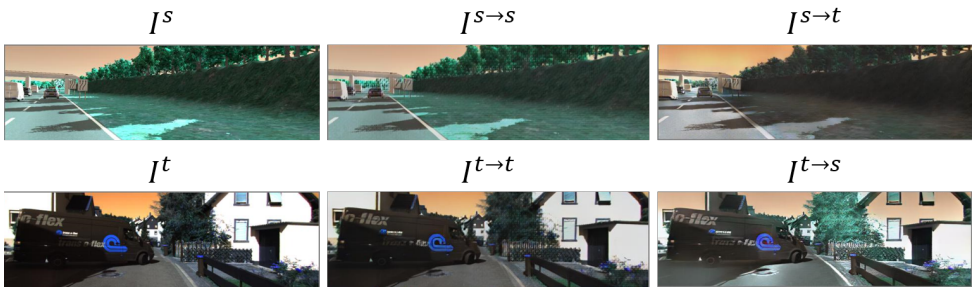

Figure 8.3: Qualitative results of the image reconstruction and translation used for feature decomposition. I^s: source input image (vKITTI), I^t: target input image (KITTI), $I^{s\to s}$: source reconstruction, $I^{t\to t}$: target reconstruction, $I^{s\to t}$: source-to-target translation, $I^{t\to s}$: target-to-source translation.

+Tgt+AL makes an obvious improvement over Src-Only, showing the importance of domain adaptation. Second, +Tgt+Con+2BN refers to the model that makes use of the decomposed content features and deploys two separate BN branches for the source and target domains, respectively. +Tgt+Con+2BN greatly improves the abs-rel metric by 0.017, showing our feature decomposition and separate BNs are effective in learning the domain-invariant content feature. Next, +Tgt+Con+2BN+Sty includes z^t_{sty} in the pipeline but still maintains only two separate BNs. Results show that it suffers from severe performance degradation. This proves our argument that content and style features have different distributions, so passing them through the same BN would drop model performance. Finally, LFDA (i.e. +Tgt+Con+3BN+Sty), which deploys the third BN for the target style feature exclusively, resolves this issue successfully. Obviously, LFDA performs the best in most metrics, demonstrating the effectiveness of our method.

8.3.6 Feature Decomposition Visualization

To verify the effectiveness of our feature decomposition, Figure 8.3 shows the qualitative results of the image reconstruction and translation that are illustrated in Figure 8.2

Table 8.6: Comparison of model complexity. The number of multiply-accumulate operations (MACs) is computed on the input size of 192×640.

Method	Params	MACs
GASDA [259]	112.3M	221.5G
SharinGAN [171]	57.7M	148.1G
LFDA (Ours)	**57.6M**	**108.1G**

(b). We can observe that the reconstructed images $I^{s\to s}$ and $I^{t\to t}$ are very close to the input images I^s and I^t, respectively. In addition, the translated images $I^{s\to t}$ and $I^{t\to s}$ accurately maintain the content information while generating the style appearance of another domain. Such high-quality results can be achieved only if the decomposition of content and style features is successful. This demonstrates the rationale behind the high performance of the proposed method.

8.3.7 Computational Complexity

In addition to accuracy, model size and computational cost are also important factors when we evaluate a model. They determine the feasibility of a model for practical applications. In Table 8.6, we compare LFDA to two existing top-performing approaches in terms of the number of parameters and the number of multiply-accumulate operations (MACs) used at inference time. GASDA [259] includes three sub-networks during inference, a target data MDE network, a target-to-source translation network, and a target-to-source MDE network. This design places a heavy computational burden. SharinGAN [171] also needs an image translation network plus a MDE network. In contrast, in LFDA, the only sub-network in addition to the primary MDE network is E^t_{sty}, which increases minimum complexity. LFDA's number of MACs is 51% and 20% fewer than GASDA and SharinGAN, respectively, showing that our method can

bridge the domain gap much more efficiently.

8.4 Summary

In this chapter, we propose LFDA, a novel domain adaptive MDE method. We suppose that a feature space can be decomposed into components of image content and appearance style. LFDA learns to achieve this decomposition and thus can efficiently mitigate the domain shift problem between source and target data. LFDA shows superior accuracy on three broad scenarios of domain adaptation. Moreover, it has a relatively low computational cost and can be trained end-to-end in a single stage, thereby more practical for real-world applications.

Chapter 9

Spatio-Temporal Pixel-Level Contrastive Learning-based Source-Free Domain Adaptation for Video Semantic Segmentation

9.1 Motivation

Under the SFDA setting [30, 121, 125, 221, 251], label supervision is not available (see details in Chapter 2). Most SFDA studies adopt pseudo-supervision or self-supervision techniques to adapt the source-trained model to the target domain [77, 197]. However, they consider only image-level information for model adaptation. In many real-world semantic segmentation applications (autonomous driving, safety surveillance, etc.), we have to deal with temporal data such as streams of images or videos. Supervised approaches that use temporal information have been successful for Video Semantic Segmentation (VSS), which predicts pixel-level semantics for each video frame [85, 106, 131, 224]. Recently, video-based UDA strategies have also been developed and yielded better performance than image-based UDA on VSS [63, 194, 241]. This motivates us to propose a novel SFDA method for VSS, leveraging temporal information to tackle the absence of source data better. In particular, we find that current image-based SFDA approaches suffer from sub-optimal performance when applied to VSS (see Figure 9.1). To the best of our knowledge, this is the first work to explore video-based SFDA solutions.

In this chapter, we propose a novel spatio-temporal SFDA method namely Spatio-Temporal Pixel-Level (STPL) Contrastive Learning (CL), which takes full advantage of both spatial and temporal information for adapting VSS models. STPL consists of two main stages. (1) Spatio-temporal feature extraction: First, given a target video sequence input, STPL fuses the RGB and optical flow modalities to extract spatio-temporal features from the video. Meanwhile, it performs cross-frame augmentation via randomized spatial transformations to generate an augmented video sequence, then extracts augmented spatio-temporal features. (2) Pixel-level contrastive learning: Next,

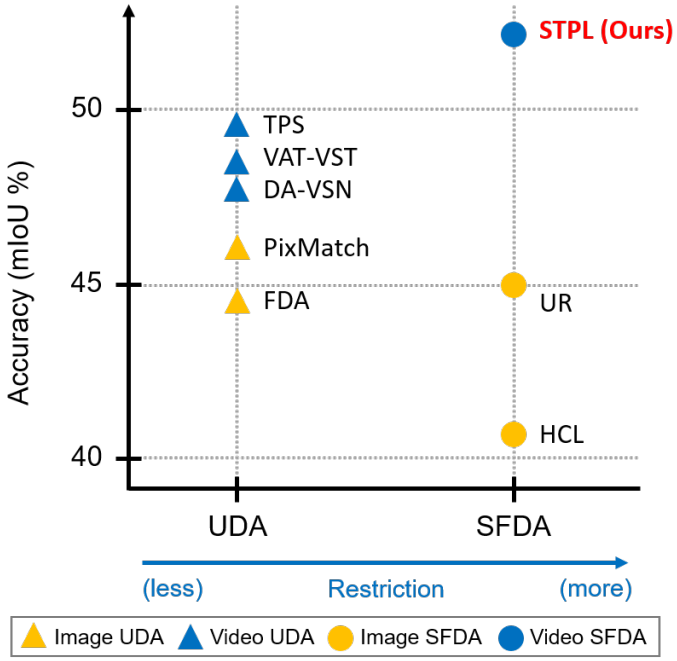

Figure 9.1: Comparison of VSS accuracy. Video-based UDA methods [63, 194, 241] outperform image-based UDA methods [156, 249], showing the importance of video-based strategies for the VSS task. Image-based SFDA methods [77, 197] perform lower than the UDA methods, which shows the difficulty of the more restricted SFDA setting. The proposed STPL, even with SFDA, achieves the best accuracy and locates at the top-right corner of the chart (i.e., more restriction, but higher accuracy).

STPL optimizes a pixel-level contrastive loss between the original and augmented spatio-temporal feature representations. This objective enforces representations to be compact for same-class pixels across both the spatial and temporal dimensions.

With these designs, STPL explicitly learns semantic correlations among pixels in the spatio-temporal space, providing strong self-supervision for adaptation to an unlabeled target domain. Furthermore, we demonstrate that STPL is a non-trivial unified spatio-temporal framework. Specifically, *Spatial-only CL* and *Temporal-only CL* are special cases of STPL, and STPL is better than a naïve combination of them. Extensive experiments demonstrate the superiority of STPL over various

baselines, including the image-based SFDA as well as image- and video-based UDA approaches that rely on source data (see Figure 9.1). Code is available at: https://github.com/shaoyuanlo/STPL

The key contributions of this chapter are summarized as follows:

- We propose a novel SFDA method for VSS. To the best of our knowledge, this is the first work to explore video-based SFDA solutions.
- We propose a novel CL method, namely STPL, which explicitly learns semantic correlations among pixels in the spatio-temporal space, providing strong self-supervision for adaptation to an unlabeled target domain.
- We conduct extensive experiments and show that STPL provides a better solution compared to the existing image-based SFDA methods as well as image- and video-based UDA methods for the given problem formulation.

9.1.1 Related Work

Video semantic segmentation. VSS predicts pixel-level semantics for each video frame [50, 76, 85, 106, 120, 131], which has been considered a crucial task for video understanding [224]. VSS networks use temporal information, the inherent nature of videos, to pursue more accurate or faster segmentation. For example, FSO [106] employs the dense conditional random field as post-processing to obtain temporally consistent segmentation. NetWarp [50] uses optical flow information to transfer intermediate feature maps of adjacent frames and gains better accuracy. ACCEL [85] integrates predictions of sequential frames via an adaptive fusion mechanism. TDNet [76] extracts feature maps across different frames and merges them by an attention

propagation module. ESVS [131] considers the temporal correlation during training and achieves a higher inference speed. These works rely on large densely annotated training data and are sensitive to domain shifts.

Unsupervised domain adaptation. Recently, there are several works studying UDA for VSS [63, 194, 241]. DA-VSN [63] presents temporal consistency regularization to minimize temporal discrepancy across different domains and video frames. VAT-VST [194] extends both adversarial learning and self-training techniques to video adaptation. TPS [241] designs temporal pseudo supervision to adapt VSS models from the perspective of consistency training. These UDA approaches rely on labeled source data for adaptation, which is not practical in many real-world scenarios.

Source-free domain adaptation. SFDA has been investigated for Image Semantic Segmentation (ISS) in recent years [77, 107, 108, 132, 197, 199]. SFDA-SS [132] develops a data-free knowledge distillation strategy for target domain adaptation. UR [197] reduces the uncertainty of target data predictions. HCL [77] presents the historical contrastive learning, which leverages the historical source hypothesis to compensate for the absence of source data. Edge/Feature-Mixup [108] generates mixup domain samples used for both source training and target adaptation. However, the need for modifying source training makes it inflexible, and it is expensive to be scaled to the video level. SFDA for videos is still relatively unexplored.

Contrastive learning. CL has been a successful representation learning technique [28, 67, 99, 98, 161]. The key idea is to create positive and negative sample pairs, then learn discriminative feature representations by maximizing the embedding distance among positive pairs and minimizing that among negative pairs. Recent works [3, 225] further explore pixel-to-pixel contrast for the ISS task, but they need label supervision

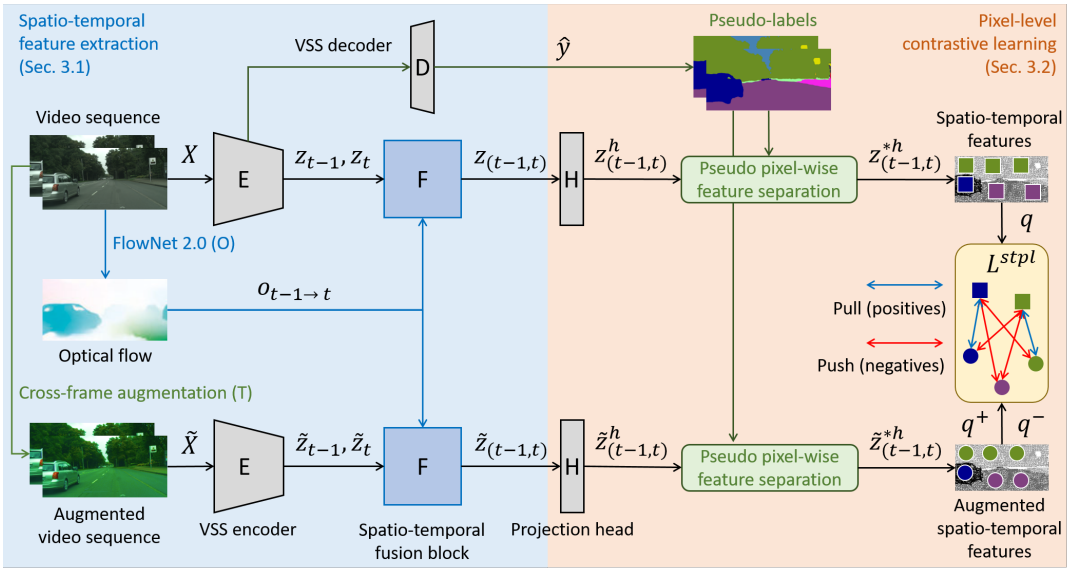

Figure 9.2: Overview of the proposed Spatio-Temporal Pixel-Level (STPL) contrastive learning framework. STPL consists of two main stages. (1) Spatio-temporal feature extraction: First, STPL fuses the RGB and optical flow ($o_{t-1\to t}$) modalities to extract spatio-temporal features ($z_{(t-1,t)}$, $\tilde{z}_{(t-1,t)}$) from both the original and augmented video sequences (X, $\tilde{X}$). (2) Pixel-level contrastive learning: Next, after passing through a projection head and pseudo pixel-wise feature separation, STPL optimizes the pixel-level contrastive loss between the original and augmented spatio-temporal features ($\mathcal{L}^{stpl}$). For simplicity, this illustration considers a two-frame video sequence as the input.

for training.

9.2 Proposed Method

An overview of the proposed STPL is illustrated in Figure 9.2. STPL is implemented by two key designs: spatio-temporal feature extraction and pixel-level CL. This section first introduces the detailed designs. Then we demonstrate that STPL is a non-trivial unified spatio-temporal framework.

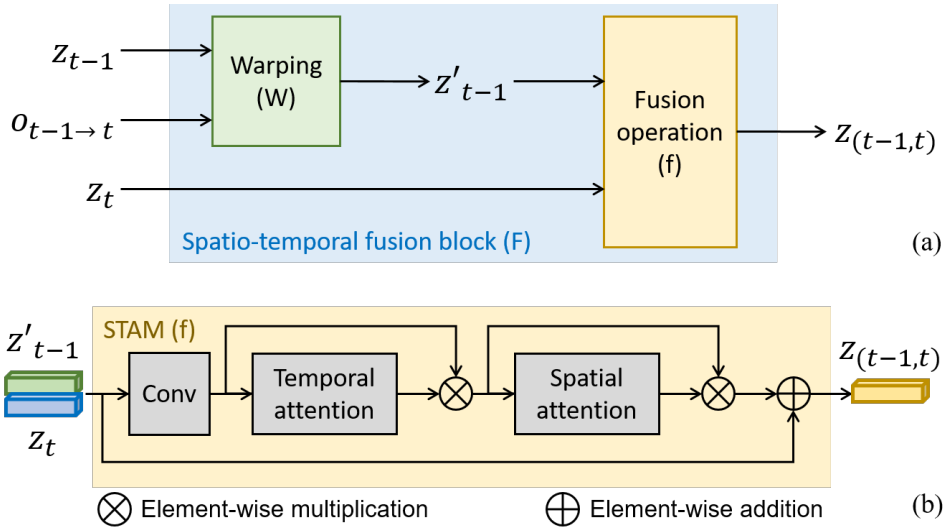

Figure 9.3: (a) The proposed spatio-temporal fusion block (F). (b) The proposed fusion operation (f): Spatio-Temporal Attention Module (STAM). STAM infers the attention of a spatio-temporal feature along the spatial and temporal dimensions separately, weighting important components in the spatio-temporal space. Details can be found in Supplementary Materials. Our fusion block is also compatible with various fusion operations.

9.2.1 Spatio-Temporal Feature Extraction

The input is an unlabeled target video sequence $X = \{x_1, x_2, ..., x_{t-1}, x_t\}$, where x_t is the current frame. For simplicity, let us consider $X = \{x_{t-1}, x_t\}$, i.e., a video with a current frame and a previous frame. Given X, the VSS network's encoder E extracts feature representations for each frame: $z_{t-1} = E(x_{t-1})$ and $z_t = E(x_t)$. In addition, we employ FlowNet 2.0 [81] denoted as O, a widely used optical flow estimator, to estimate the optical flow between the previous and the current frames as: $o_{t-1\to t} = O(x_{t-1}, x_t)$.

Spatio-temporal fusion block. Next, we propose a spatio-temporal fusion block F to extract spatio-temporal feature representations from the previous and the current features z_{t-1} and z_t (see Figure 9.3 (a)). It adopts the estimated optical flow $o_{t-1\to t}$ to warp the previous feature z_{t-1} to the propagated feature as: $z'_{t-1} = W(z_{t-1}; o_{t-1\to t})$,

where W denotes the warping operation. This feature propagation aligns the pixel correspondence between the previous and the current features, which is crucial for the dense prediction task. Then a fusion operation f is used to fuse the cross-frame features into a spatio-temporal feature as: $z_{(t-1,t)} = f(z'_{t-1}, z_t)$.

The fusion operation integrates two input features into one output feature. It can be element-wise addition, concatenation, 1×1 convolution layer, an attention module, or other variants. Inspired by [230], we design a Spatio-Temporal Attention Module (STAM) illustrated in Figure 9.3 (b). STAM infers the attention of a spatio-temporal feature along the spatial and temporal dimensions separately, weighting important components in the spatio-temporal space. Consider the concatenation of the propagated previous feature z'_{t-1} and the current feature z_t as $z'_{(t-1,t)} \in \mathbb{R}^{T\times C\times H\times W}$, the STAM process can be written as:

$$\begin{aligned} z_{(t-1,t)} = \{[A_{spa}[A_{tem}(z'_{(t-1,t)}) \otimes z'_{(t-1,t)}] \\ \otimes [A_{tem}(z'_{(t-1,t)}) \otimes z'_{(t-1,t)}]\} \oplus z'_{(t-1,t)}, \end{aligned} \tag{9.1}$$

where A_{tem} is temporal attension, A_{spa} is spatial attension, $\otimes$ denotes element-wise multiplication, and $\oplus$ denotes element-wise addition.

The proposed temporal attention mechanism learns to choose informative temporal elements along each pixel's temporal dimension in the spatio-temporal space. The temporal attention $A_{tem} \in \mathbb{R}^{T\times1\times1\times1}$ is performed as:

$$A_{tem}(z) = \sigma(FC(AvgPool(z)) + FC(MaxPool(z))), \tag{9.2}$$

where σ is the sigmoid function, and FC denotes a fully connected layer.

The spatial attention mechanism chooses informative pixels along the spatial

dimension in the spatio-temporal space. The spatial attention $A_{spa} \in \mathbb{R}^{1\times1\times H\times W}$ is performed as:

$$A_{spa}(z) = \sigma(Conv(Concat[AvgPool(z), MaxPool(z)])), \tag{9.3}$$

where $Concat$ denotes the concatenation operation, and $Conv$ denotes a convolutional layer.

Note that the main contribution of this work is the STPL framework. In Table 9.4, we can see that STPL can outperform all the existing methods even with the very simple Concatenation fusion, showing its flexibility. We propose STAM to show that STPL can further benefit from a more advanced fusion module.

Cross-frame augmentation. Meanwhile, we perform cross-frame augmentation [241] that applies randomized spatial transformations T on each input frame to generate an augmented video sequence: $\tilde{X} = T(X) = \{\tilde{x}_{t-1}, \tilde{x}_t\}$. Then we apply the same spatio-temporal feature extraction process on $\tilde{X}$ and extract the augmented spatio-temporal feature $\tilde{z}_{(t-1,t)}$. The augmentation T contains randomized Gaussian blurring and color jittering transformations.

9.2.2 Pixel-Level Contrastive Learning

With the extracted original and augmented spatio-temporal features $z_{(t-1,t)}$ and $\tilde{z}_{(t-1,t)}$, we propose a new CL method to derive a semantically meaningful self-supervision. Typical CL schemes [28, 98] assume that an input contains only a single semantic category, and needs a large batch size to offer sufficient positive/negative pairs for training. Nevertheless, in VSS, the input contains multiple instances, and a large batch size is computationally infeasible. Hence, we propose a method based on

a pixel-level CL paradigm that leverages pixel-to-pixel contrast [3, 225], and refer to our method as Spatio-Temporal Pixel-Level (STPL) CL.

Pseudo pixel-wise feature separation. STPL aims to acquire pixel-level representations that are similar among the same-class pixel samples but distinct among different-class pixel samples. Since we do not have target domain labels, we use our VSS model's prediction for the input X as pseudo-label $\hat{y}$. Subsequently, we use $\hat{y}$ to do pixel-wise feature separation. To maintain high-quality pseudo-labels, we set a hyperparameter of confident proportion k to control the proportion of pixels preserved as pseudo-labels. More precisely, the confident pseudo-labels $\hat{y}^*$ are obtained by $\hat{y}^* = topk(\hat{y}; k) \subset \hat{y}$, where $topk$ is an operation that returns the k-proportion of the most confident predictions according to their probability scores.

Pixel-to-pixel contrastive loss. To perform CL, we first adopt a projection head H to project our feature representations $z^h_{(t-1,t)} = H(z_{(t-1,t)})$ and $\tilde{z}^h_{(t-1,t)} = H(\tilde{z}_{(t-1,t)})$, similar to SimCLR [28]. According to the generated confident pseudo-labels $\hat{y}^*$, we denote the confident pixel representation sets in $z^h_{(t-1,t)}$ and $\tilde{z}^h_{(t-1,t)}$ as $z^{*h}_{(t-1,t)} \subset z^h_{(t-1,t)}$ and $\tilde{z}^{*h}_{(t-1,t)} \subset \tilde{z}^h_{(t-1,t)}$, respectively. Next, consider a query confident pixel representation $q \in z^{*h}_{(t-1,t)}$ (i.e., q is a pixel representation in the feature $z^{*h}_{(t-1,t)}$) with a predicted pseudo-label $\hat{y}^*_q$, we define its positive pair set as:

$$P_q \equiv \{q^+ \in \tilde{z}^{*h}_{(t-1,t)} : \hat{y}^*_{q^+} = \hat{y}^*_q\}, \tag{9.4}$$

i.e., all the same-class pixels in the augmented feature $\tilde{z}^{*h}_{(t-1,t)}$. Then we define its negative pair set as:

$$N_q \equiv \{q^- \in \tilde{z}^{*h}_{(t-1,t)} : \hat{y}^*_{q^-} \neq \hat{y}^*_q\}, \tag{9.5}$$

i.e., all the different-class pixels in $\tilde{z}^{*h}_{(t-1,t)}$. We follow SupCon [98] to develop a CL

scheme with multiple positive pairs. The complete formulation of the proposed STPL contrastive loss is as follows:

$$\mathcal{L}_q^{stpl} = \frac{-1}{|P_q|} \sum_{q^+ \in P_q} \log \frac{\exp(q \cdot q^+ / \tau)}{\sum_{q^- \in N_q} \exp(q \cdot q^- / \tau)}, \tag{9.6}$$

where τ is a temperature parameter, and the $\cdot$ symbol denotes the inner product. Finally, the overall objective for the given video sequence input X is defined as:

$$\mathcal{L}^{stpl} = \frac{1}{|z^{*h}_{(t-1,t)}|} \sum\nolimits_{q \in z^{*h}_{(t-1,t)}} \mathcal{L}_q^{stpl}. \tag{9.7}$$

This objective enforces the pixel representations in the original spatio-temporal features to be similar to that of the same-class pixels in the augmented features, while being distinct from that of the different-class pixels. This explicitly learns semantic correlations among pixels in the spatio-temporal space and thus can achieve better class discriminability. The proposed STPL provides a strong self-supervision for video adaptation under the SFDA setup.

9.2.3 STPL as a Unified Spatio-Temporal Framework

We further demonstrate that STPL is a non-trivial unified spatio-temporal framework. Specifically, *Spatial-only CL* and *Temporal-only CL* are special cases of STPL. Moreover, we show that a naïve combination of them is sub-optimal compared to STPL.

Spatial-only contrast. Let us turn off the fusion operation F of the STPL framework with an identity operation. Then, let us allow only the current frame feature z_t, and similarly, only the augmented current frame feature $\tilde{z}_t$ to pass through the fusion block. After the projection head and confident filtering steps, the contrastive loss

would be computed between z_t^{*h} and $\tilde{z}_t^{*h}$ instead of the spatio-temporal features $z_{(t-1,t)}^{*h}$ and $\tilde{z}_{(t-1,t)}^{*h}$. That is, in Eq. (9.6) and Eq. (9.7), it becomes that $q \in z_t^{*h}$ and $\{q^+, q^-\} \in \tilde{z}_t^{*h}$. This computes contrast between only spatial variations and thus is a spatial-only special case of STPL. We denote this loss as $\mathcal{L}^{spa}$.

Temporal-only contrast. Let us consider a duplicate copy of the input video as an augmentation (i.e., $\tilde{X} = X$). Next, let us turn off the fusion operation F of STPL, allowing only the current frame feature z_t and the augmented previous frame feature $\tilde{z}_{t-1}$ to pass through the fusion block. Here $\tilde{z}_{t-1} = z_{t-1}$ since $\tilde{X} = X$. Hence, after the projection head and confident filtering steps, the contrastive loss would be computed between z_t^{*h} and z_{t-1}^{*h}. That is, in Eq. (9.6) and Eq. (9.7), it becomes that $q \in z_t^{*h}$ and $\{q^+, q^-\} \in z_{t-1}^{*h}$. This computes contrast between only temporal variations and thus is a temporal-only special case of STPL. We denote this loss as $\mathcal{L}^{tem}$.

Naïve combination. To learn spatio-temporal contrast, a naïve way would be to combine the spatial-only and temporal-only contrastive losses together: $\mathcal{L}^{spa} + \mathcal{L}^{tem}$. Our experiments in Sec. 9.3.3 show that the naïve combination is sub-optimal compared to STPL. This demonstrates that the proposed STPL is a non-trivial unified spatio-temporal framework. Figure 9.4 compares the proposed spatio-temporal contrast $\mathcal{L}^{stpl}$, spatial-only contrast $\mathcal{L}^{spa}$, and temporal-only contrast $\mathcal{L}^{tem}$.

9.3 Experiments

9.3.1 Experimental Setup

Datasets. We evaluate our method on two widely used domain adaptive VSS benchmarks: VIPER [176] $\rightarrow$ Cityscapes-Seq [32] and SYNTHIA-Seq [178] $\rightarrow$

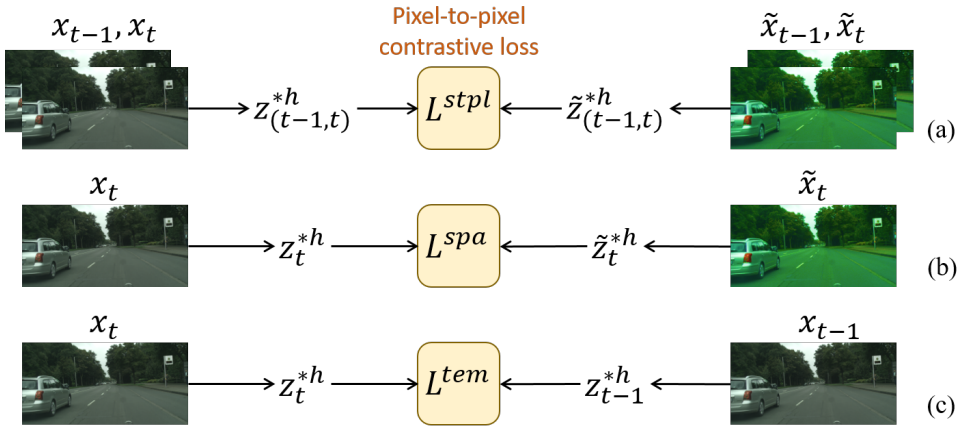

Figure 9.4: Illustration of (a) the proposed spatio-temporal contrast $\mathcal{L}^{stpl}$ (Eq. (9.6), (9.7)), (b) spatial-only contrast $\mathcal{L}^{spa}$, and (c) temporal-only contrast $\mathcal{L}^{tem}$.

Cityscapes-Seq. VIPER has 133,670 synthetic video frames with a resolution of 1080×1920. SYNTHIA-Seq consists of 8,000 synthetic video frames with a resolution of 760×1280. We consider VIPER and Synthia-Seq as source datasets to pre-train source models, respectively. Cityscapes-Seq is a realistic traffic scene dataset. It contains 2,975 training and 500 validation video sequences with a frame resolution of 1024×2048. We use it as a target dataset. Following [63, 241], we resize the frames of VIPER and Cityscapes-Seq to 760×1280 and 512×1024, respectively. For evaluations, the output predictions are interpolated to the original size.

Implementation details. Following [63, 241], we employ ACCEL [85] as our VSS network. It includes two segmentation branches, an optical flow estimation branch, and a prediction fusion layer. These branches consist of DeepLabv2 [25] architecture with ResNet-101 [69] backbone, FlowNet [39], and a 1×1 convolution layer, respectively. All the adaptation models are trained by an SGD optimizer with an initial learning rate of $2.5e^{-6}$ and a momentum of 0.9 for 20k iterations. The learning rate decreases along the polynomial decay with a power of 0.9. We set the temperature $\tau = 0.07$

Table 9.1: Quantitative comparisons (%) with multiple types of domain adaptation baselines on VIPER → Cityscapes-Seq.

Method	Design	DA	road	side.	buil.	fence	light	sign	vege.	terr.	sky	pers.	car	truck	bus	mot.	bike	mIoU
Source-only	-	-	56.7	18.7	78.7	6.0	22.0	15.6	81.6	18.3	80.4	59.9	66.3	4.5	16.8	20.4	10.3	37.1
FDA [249]	Image	UDA	70.3	27.7	81.3	17.6	25.8	20.0	83.7	31.3	82.9	57.1	72.2	22.4	49.0	17.2	7.5	44.4
PixMatch [156]	Image	UDA	79.4	26.1	84.6	16.6	28.7	23.0	85.0	30.1	83.7	58.6	75.8	34.2	45.7	16.6	12.4	46.7
RDA [78]	Image	UDA	70.3	27.7	81.3	17.6	25.8	20.0	83.7	31.3	82.9	57.1	72.2	22.4	49.0	17.2	7.5	44.4
UR [197]	Image	SFDA	84.2	20.1	80.1	11.5	30.7	31.1	82.8	22.1	69.2	59.5	81.0	4.9	52.7	36.6	8.7	45.0
HCL [77]	Image	SFDA	80.6	34.0	76.8	29.7	20.5	36.3	79.1	19.2	56.3	58.1	73.9	3.4	5.2	20.0	28.9	41.5
DA-VSN [63]	Video	UDA	86.8	36.7	83.5	22.9	30.2	27.7	83.6	26.7	80.3	60.0	79.1	20.3	47.2	21.2	11.4	47.8
VAT-VST [194]	Video	UDA	87.1	41.2	82.2	17.1	26.0	33.1	83.2	20.6	70.6	64.3	71.0	11.6	84.1	27.8	11.1	48.7
TPS [241]	Video	UDA	82.4	36.9	79.5	9.0	26.3	29.4	78.5	28.2	81.8	61.2	80.2	39.8	40.3	28.5	31.7	48.9
DA-VSN* [63]	Video	SFDA	77.8	32.6	79.6	29.2	37.5	34.7	82.0	22.0	64.1	61.1	76.0	6.6	32.8	32.2	11.4	45.3
VAT-VST* [194]	Video	SFDA	48.2	20.4	78.1	28.8	33.1	33.6	81.1	20.0	56.1	58.3	74.7	8.6	73.5	29.7	9.6	43.6
TPS* [241]	Video	SFDA	69.9	0.0	77.4	0.0	6.2	14.8	77.5	0.2	47.4	36.9	67.7	0.0	19.3	0.0	0.0	27.8
STPL (Ours)	Video	SFDA	83.1	38.9	81.9	48.7	32.7	37.3	84.4	23.1	64.4	62.0	82.1	20.0	76.4	40.4	12.8	52.5
Oracle	-	-	96.5	76.8	89.2	58.3	49.5	60.0	90.3	37.5	80.5	72.1	92.0	41.6	64.6	63.1	76.2	69.9

and the confident proportion $k = 0.7$. The mean Intersection-over-Union (mIoU) is used as the evaluation metric. Our experiments are implemented using PyTorch [166].

9.3.2 Main Results

Baselines. Since the proposed STPL is the first SFDA method for VSS, we compare it with multiple related domain adaptation state-of-the-art approaches described as follows. (1) *Image-based UDA*: FDA [249], PixMatch [156] and RDA [78]; (2) *Image-based SFDA*: UR [197] and HCL [77]; and (3) *Video-based UDA*: DA-VSN [63], VAT-VST [194] and TPS [241]. The image-based approaches are applied to videos by using a VSS backbone (ACCEL in our experiments), following the practice of [63, 241]. Furthermore, to fairly assess our STPL, we create the SFDA versions of these video-based UDA approaches as our (4) *Video-based SFDA* baselines. We remove all of their loss terms containing source data while keeping all the loss terms

computed from only target data. We use the * symbol to denote these baselines. The results of the source-only and oracle (i.e., trained with target domain labels) models are also reported for reference. For fair comparisons, all four types of baselines use the same VSS backbone and training settings.

VIPER → Cityscapes-Seq. Table 9.1 reports the evaluation results on the VIPER → Cityscapes-Seq adaptation benchmark. The proposed STPL outperforms all four types of baselines by decent margins, which is 15.1% higher than the source-only model and 3.6% higher than the best-performing competitor. In particular, its superiority over the image-based SFDA approaches indicates the benefits of a video-based solution and demonstrates the effectiveness of our spatio-temporal strategy for videos. We can also observe that the video-based UDA approaches suffer from performance degradation when applied to SFDA. Whereas, STPL achieves better performance even compared to their UDA results relying on source data.

SYNTHIA-seq → Cityscapes-Seq. Table 9.2 provides the results on the SYNTHIA-Seq → Cityscapes-Seq benchmark. Similarly, our STPL is better than most baselines. Although TPS achieves the best accuracy under UDA, this requires accessing source data. Moreover, TPS*'s accuracy dramatically reduces to 22.1% under SFDA, showing that it is not a proper solution when source data are unavailable. Overall, these results clearly demonstrate the superiority of STPL.

Qualitative results. Figure 9.5 shows examples of qualitative results on VIPER → Cityscapes-Seq. The source-only model produces noisy and inconsistent predictions on the road and sidewalk, showing the domain shift effect. UR, an image-based SFDA method, suffers from inaccurate sky predictions and cannot detect the whole sidewalk. In contrast, the proposed STPL obtains more accurate segmentation results with high

Table 9.2: Quantitative comparisons (%) with multiple types of domain adaptation baselines on SYNTHIA-Seq → Cityscapes-Seq.

Method	Design	DA	road	side.	buil.	pole	light	sign	vege.	sky	pers.	rider	car	mIoU
Source-only	-	-	56.3	26.6	75.6	25.5	5.7	15.6	71.0	58.5	41.7	17.1	27.9	38.3
FDA [249]	Image	UDA	84.1	32.8	67.6	28.1	5.5	20.3	61.1	64.8	43.1	19.0	70.6	45.2
PixMatch [156]	Image	UDA	90.2	49.9	75.1	23.1	17.4	34.2	67.1	49.9	55.8	14.0	84.3	51.0
RDA [78]	Image	UDA	84.7	26.4	73.9	23.8	7.1	18.6	66.7	68.0	48.6	9.3	68.8	45.1
UR [197]	Image	SFDA	83.5	8.0	68.1	16.5	9.9	17.7	62.4	65.1	31.9	15.3	82.3	41.9
HCL [77]	Image	SFDA	79.0	44.7	78.9	25.4	12.9	36.6	75.2	63.0	49.0	19.5	50.1	48.6
DA-VSN [63]	Video	UDA	89.4	31.0	77.4	26.1	9.1	20.4	75.4	74.6	42.9	16.1	82.4	49.5
VAT-VST [194]	Video	UDA	82.8	26.5	78.3	23.7	12.8	20.0	78.4	64.5	45.5	16.0	69.6	47.1
TPS [241]	Video	UDA	91.2	53.7	74.9	24.6	17.9	39.3	68.1	59.7	57.2	20.3	84.5	53.8
DA-VSN* [63]	Video	SFDA	81.0	37.9	68.4	23.7	14.0	27.5	69.8	71.3	46.4	18.7	80.2	49.0
VAT-VST* [194]	Video	SFDA	84.8	28.6	72.4	25.6	17.1	32.9	64.5	56.9	50.7	21.9	83.4	49.0
TPS* [241]	Video	SFDA	62.6	0.0	69.2	0.2	0.8	14.4	56.6	10.4	4.2	0.2	24.5	22.1
STPL (Ours)	Video	SFDA	87.6	42.5	74.6	27.7	18.5	35.9	69.0	55.5	54.5	17.5	85.9	51.8
Oracle	-	-	96.4	78.1	89.1	43.6	42.3	64.9	90.3	84.4	66.8	50.7	92.7	72.7

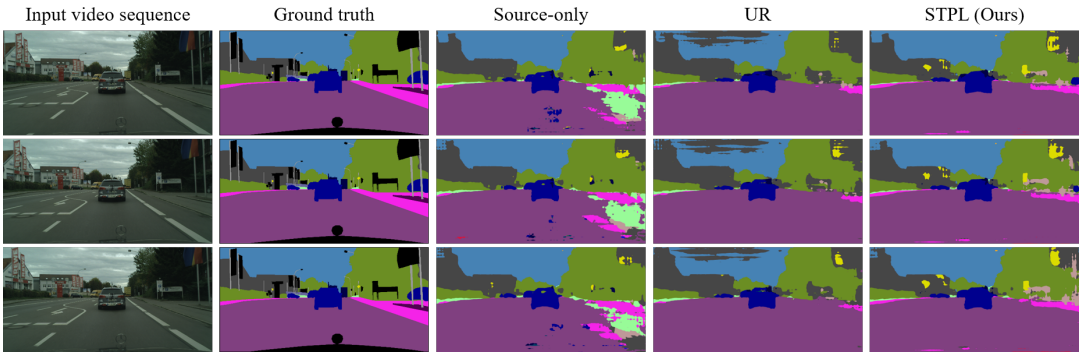

Figure 9.5: Qualitative results on VIPER → Cityscapes-Seq. The source-only model produces noisy and inconsistent predictions on the road and sidewalk. UR [197], an image-based SFDA method, suffers from inaccurate predictions on the sky and sidewalk. In contrast, the proposed STPL obtains more accurate segmentation results with high temporal consistency across the video sequence.

temporal consistency across the video sequence. This indicates the importance of a video-based strategy for the VSS task and demonstrates our method's effectiveness. The qualitative and quantitative results are consistent.

Table 9.3: Ablation study of different objective functions on VIPER → Cityscapes-Seq.

Method / Objective function	mIoU
Source-only	37.1
Vanilla Self-training	45.4 (+8.3)
Duplicate CL	45.7 (+8.6)
Temporal-only CL ($\mathcal{L}^{tem}$)	47.4 (+10.3)
Spatial-only CL ($\mathcal{L}^{spa}$)	51.1 (+14.0)
Naïve T+S CL ($\mathcal{L}^{tem} + \mathcal{L}^{spa}$)	51.4 (+14.3)
STPL (Ours; $\mathcal{L}^{stpl}$)	**52.5 (+15.4)**

9.3.3 Ablation Analysis

Objective functions. We conduct an ablation study to validate the effectiveness of our spatio-temporal objective for adaptation. We create several variants for comparison. *Vanilla Self-training* simply computes the cross-entropy loss between predictions and pseudo-labels with a confident threshold. *Duplicate CL* computes the pixel-level contrastive loss between two identical video frames, i.e., the loss described in Sec. 9.2.2 but uses a duplicate copy as an augmentation and passes through the current frame features only. *Temporal-only CL*, *Spatial-only CL* and *Naïve T+S CL* are described in Sec. 9.2.3, whose objective functions are $\mathcal{L}^{tem}$, $\mathcal{L}^{spa}$ and $\mathcal{L}^{tem} + \mathcal{L}^{spa}$, respectively.

As can be seen in Table 9.3, the simple Duplicate CL achieves higher accuracy than Vanilla Self-training, showing the effectiveness of the pixel-level contrastive loss. Both Temporal-only CL and Spatial-only CL make an improvement over Duplicate CL, which indicates the importance of contrasting with variations. Naïve T+S CL, a naïve combination of the temporal-only and spatial-only contrastive losses, is slightly better than either single loss. The proposed spatio-temporal objective further outperforms Naïve T+S CL, showing that our design can learn more semantically

Table 9.4: Ablation study of different fusion operations f on VIPER $\rightarrow$ Cityscapes-Seq.

Fusion operation	mIoU
Element-wise addition	51.4
1×1 convolution layer	51.8
Concatenation	52.3
STAM	**52.5**

Table 9.5: Temporal consistency of different objective functions on VIPER $\rightarrow$ Cityscapes-Seq.

Method / Objective function	Consistency (%)
Source-only	72.93
Temporal-only CL ($\mathcal{L}^{tem}$)	75.84 (+2.91)
Spatial-only CL ($\mathcal{L}^{spa}$)	77.68 (+4.75)
Naïve T+S CL ($\mathcal{L}^{tem} + \mathcal{L}^{spa}$)	80.91 (+7.89)
STPL (Ours; $\mathcal{L}^{stpl}$)	**82.14 (+9.21)**

meaningful context from the spatio-temporal space than simply adding the losses of two dimensions together. This demonstrates that our STPL is a non-trivial unified spatio-temporal framework for video adaptation.

Fusion operations. As discussed in Sec. 9.2.1, our STPL framework is compatible with various fusion operations used to extract spatio-temporal features. Here we consider and compare different fusion operations, such as element-wise addition, 1×1 convolution layer, concatenation, and the proposed STAM module. In Table 9.4, we can observe that STAM achieves the best performance, showing its effectiveness. On the other hand, adopting any fusion operation can outperform all the baselines in Table 9.1 and variants in Table 9.3. This demonstrates that STPL maintains superior performance regardless of the choice of fusion operations.

Temporal consistency. We quantitatively compare the temporal consistency of different objective functions. The temporal consistency is derived from the overlap

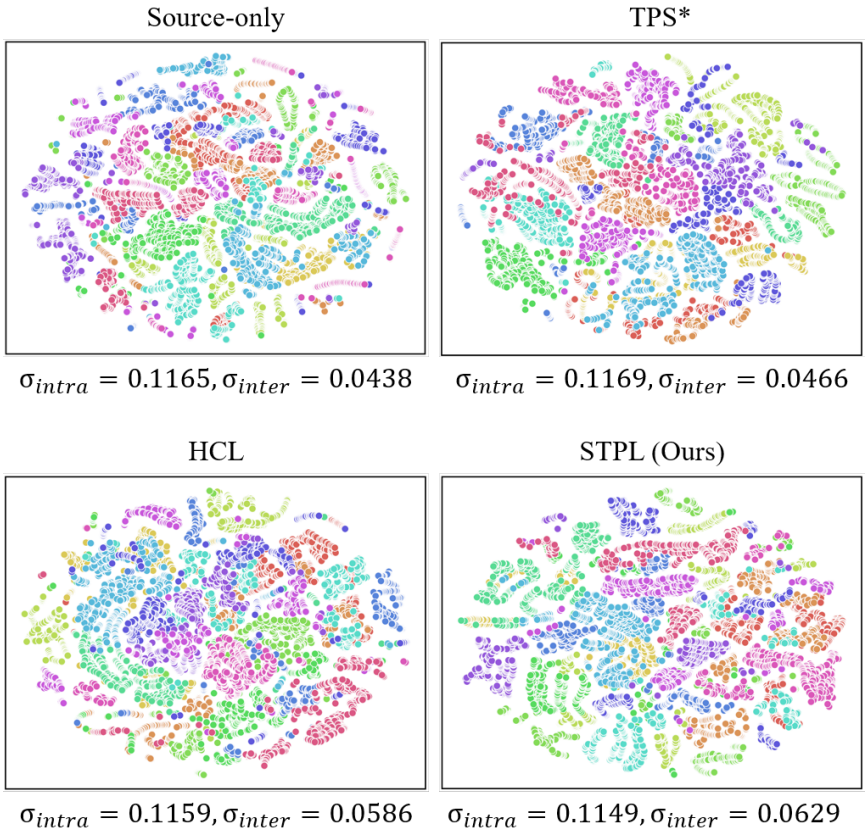

Figure 9.6: The t-SNE visualization [216] of the feature space learned for VIPER → Cityscapes-Seq, where each point in the scatter plots stands for a pixel representation. σ_{intra} is the intra-class variance (lower is better) and σ_{inter} is the inter-class variance (higher is better) of the feature space. All the methods are evaluated on the same selected video samples. In comparison, the proposed STPL learns the most discriminative feature space, which is reflected by the lowest σ_{intra} and the highest σ_{inter}.

between the predicted segmentation maps of successive frames. We compute the percentage of overlapping pixels. As shown in Table 9.5, STPL performs the best, indicating that the proposed spatio-temporal method significantly improves temporal consistency. This quantitative result is consistent with the qualitative results shown in Figure 9.5.

Feature visualization. Figure 9.6 provides the t-SNE visualization [216] of the feature space learned for the VIPER → Cityscapes-Seq benchmark. Each point

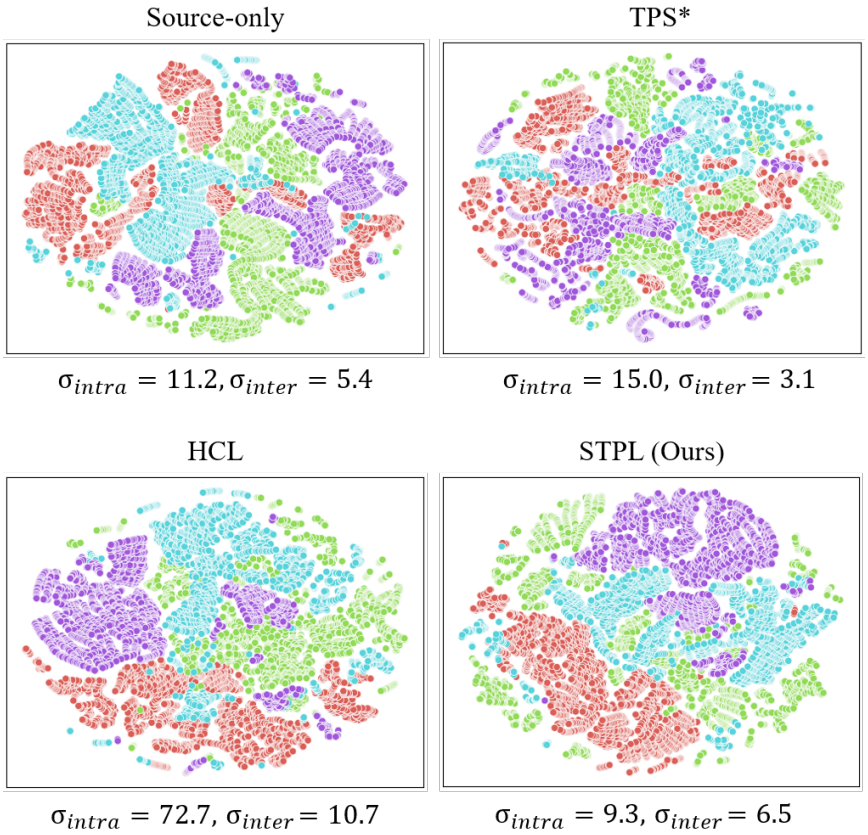

Figure 9.7: The t-SNE visualization [216] of the feature space learned for VIPER $\rightarrow$ Cityscapes-Seq, where each point in the scatter plots stands for a pixel representation. Four classes (road, traffic light, car, and bicycle) are sampled to visualize. The proposed STPL learns the most discriminative feature space, which is reflected by the lowest σ_{intra} and the high σ_{inter}.

in the scatter plots stands for a pixel representation. We compute the intra-class variance σ_{intra} (lower is better) and inter-class variance σ_{inter} (higher is better) of the feature space to provide a quantitative measurement. As can be seen, TPS*, which is originally designed for UDA, has a less discriminative feature space under the SFDA setup. It obtains higher σ_{intra} than the source-trained model. HCL, an image-based SFDA approach, has a relatively better feature space. It acquires lower σ_{intra} and higher σ_{inter}. In comparison, the proposed STPL learns the most discriminative

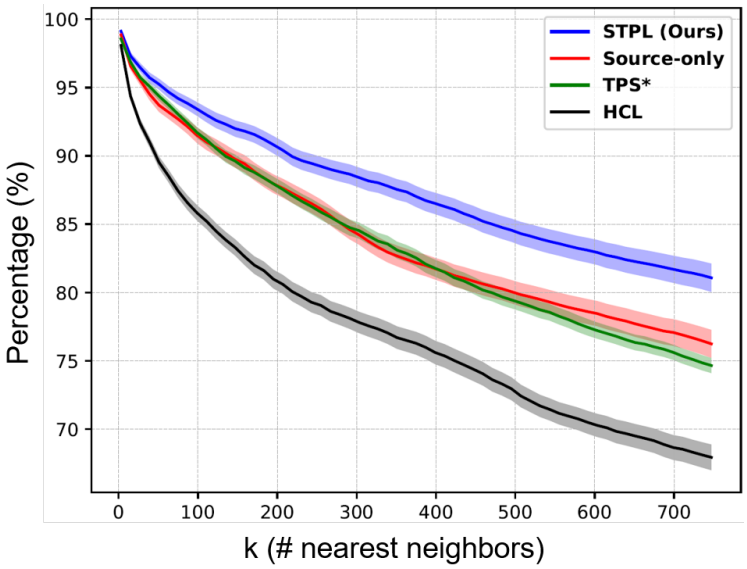

Figure 9.8: The percentage of same-class pixel representations among the k-nearest neighbors in the feature space. STPL achieves higher percentage for every k value, showing that STPL learns a more discriminative and semantically consistent feature space.

feature space. Unlike HCL, STPL leverages spatio-temporal information for video adaptation, and the benefit is clearly reflected by the lowest σ_{intra} and the highest σ_{inter}. This demonstrates STPL's ability to learn semantic correlations among pixels in the spatio-temporal space.

For more clear observations, we provide another t-SNE visualization Figure 9.7 by sampling only four classes (road, traffic light, car, and bicycle). Similarly, TPS* obtains a higher σ_{intra} and a lower σ_{inter} than the source-trained model. HCL acquires a higher σ_{inter}, but its σ_{intra} is much higher. In comparison, STPL has the lowest σ_{intra} and the high σ_{inter}. This one again demonstrates STPL's superiority.

Feature space neighborhood. This analysis inspects the neighborhood of the feature space learned by the proposed STPL, which quantitatively measures the discriminability of a feature space [246]. We randomly select several video samples and extract

the features at the pixel level. For an unbiased analysis, 500 pixel representations are considered for each semantic class to create a feature analysis set. Next, we query each representation in the set and retrieve the k-nearest neighbors of that representation. Among the retrieved k nearest representations, we inspect the percentage of the same-class representations it contains.

Figure 9.8 reports the inspection results. For smaller k values, all the methods have similar accuracy, which indicates that their feature spaces have semantically consistent neighbors for query pixel representations. Interestingly, when we increase the k values to retrieve more neighbors, the accuracy differences between the proposed STPL and the other approaches significantly enlarge. In other words, the accuracy of STPL drops much slower than the rest. We can see that for any given k values, STPL has more semantically consistent representations in the neighborhood. This analysis shows that the proposed method can effectively learn a discriminative feature space, thereby resulting in better performance.

9.4 Summary

In this chapter, we propose STPL, a novel SFDA method for VSS, which takes full advantage of spatio-temporal information to tackle the absence of source data better. STPL explicitly learns semantic correlations among pixels in the spatio-temporal space and provides strong self-supervision for video adaptation. To the best of our knowledge, this is the first work to explore video-based SFDA solutions. Moreover, we demonstrate that STPL is a non-trivial unified spatio-temporal framework. Extensive experiments show the superiority of STPL over various baselines, including the image-based SFDA as well as image- and video-based UDA approaches. Further insights

into the proposed method are also provided by our comprehensive ablation analysis.

Limitations. Similar to all the existing SFDA methods, the proposed STPL assumes that the source-trained model has learned source knowledge well. A sub-optimal source-trained model would affect adaptation performance. Such limitation of SFDA is an interesting direction for future investigations.

Potential negative social impact. The proposed method may make attackers easier to adapt pre-trained open-source models to malicious uses. To avoid such risk, computer security or defense mechanisms could be incorporated.

Part III

Intersection of Adversarial Robustness and Domain Adaptation

Chapter 10

Exploring Adversarially Robust Training for Unsupervised Domain Adaptation

10.1 Motivation

Although recent UDA approaches achieve impressive performance [52, 53, 145, 146, 147, 210], they do not consider the robustness against adversarial attacks [13, 201], which causes critical concerns. Currently, AT-based defenses [60, 97, 150, 256] have been considered the most effective, especially under the white-box setting [5]. Nevertheless, conventional AT requires ground-truth labels to generate adversarial examples (see details in Chapter 2). This makes it not applicable to the UDA problem since UDA considers the scenario that label information is unavailable to a target domain. A nearly contemporary work [6] resorts to external adversarially pre-trained ImageNet [35] models as teacher models to distill robustness knowledge. However, its performance is highly sensitive to the teacher models' perturbation budget, architecture, etc., which limits the flexibility in a wide range of uses. Another very recent work [245] uses an external pre-trained UDA model to produce pseudo labels for doing AT on target data. Unfortunately, we show that it suffers from suboptimal accuracy and robustness against white-box attacks.

Given the above observations, intuitive questions emerge: *Can we develop an AT algorithm specifically for the UDA problem? How to improve the unlabeled data robustness via AT while learning domain-invariant features for UDA?* In this chapter, we seek to answer these questions by systematically studying multiple AT variants that can potentially be applied to UDA. First, we apply a conventional AT [150] to an UDA model to see its effectiveness. In other words, the AT is performed on only the labeled source data. Second, inspired by [97, 256], we attempt to train models by minimizing the difference between the output logits of clean target data and the corresponding adversarial examples. With this, we can conduct a kind of AT directly

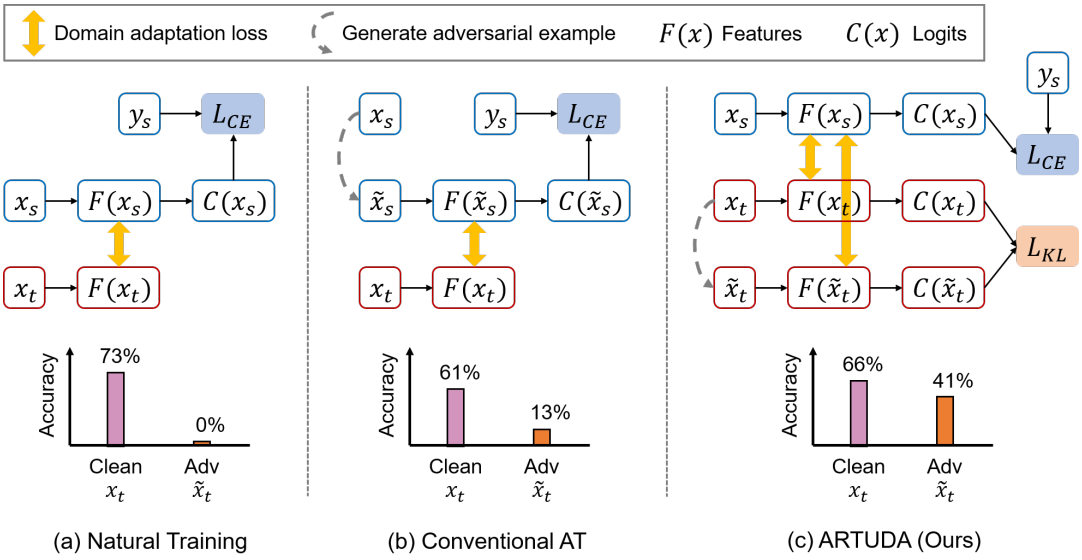

Figure 10.1: Overview of the proposed ARTUDA and its importance. L_{CE}: Cross-entropy loss. L_{KL}: KL divergence loss. Compared to conventional AT [150], ARTUDA significantly improves adversarial robustness while maintaining decent clean accuracy. We use DANN [53] with ResNet-50 [69] backbone, the VisDA-2017 [167] dataset, and the PGD-20 [150] attack for this experiment.

on the target data in a self-supervised manner. We call it *Self-Supervised Adversarial Training* or *Self-Supervised AT*. Next, we look into the effects of clean images and adversarial examples in the AT for UDA. We present the trade-off behind different AT variants. Last, we observe that BN [82] plays an important role in the AT for UDA. The feature statistic estimations at training time would affect an UDA model's robustness.

Through these investigations, we propose a novel Adversarially Robust Training method for UDA accordingly, referred to as Adversarially Robust Training for UDA (ARTUDA). It uses both source and target data for training and does not require target domain labels, so it is feasible for UDA. Moreover, it does not need guidance from external models such as adversarially pre-trained models and pre-trained UDA models. Figure 10.1 illustrates an overview and the importance of the proposed ARTUDA.

The naturally trained (i.e., train with only clean data) model's accuracy decreases to 0% under an adversarial attack. Conventional AT [150] improves robust accuracy to 13% but sacrifices clean accuracy. As can be seen, ARTUDA significantly increases robust accuracy to 41% while maintaining better clean accuracy. This shows that our method can improve unlabeled data robustness and learn domain-invariant features simultaneously for UDA. To the best of our knowledge, ARTUDA is the first AT-based UDA defense that is robust against white-box attacks. In Sec. 10.3, we extensively evaluate ARTUDA on five adversarial attacks, three datasets and three different UDA algorithms. The results demonstrate its wide range of effectiveness. Code is available at: https://github.com/shaoyuanlo/ARTUDA

The main contributions of this chapter are summarized as follows:

- We provide a systematic study into various AT methods that are suitable for UDA. We believe that such experimental analysis would provide useful insight into this relatively unexplored research direction.
- We propose ARTUDA, a new AT method specifically designed for UDA. To the best of our knowledge, it is the first AT-based UDA defense method that is robust against white-box attacks.
- Comprehensive experiments show that ARTUDA consistently improves UDA models' adversarial robustness under multiple attacks and datasets.

10.1.1 Related Work

RFA [6] and ASSUDA [245] are the most related works in the literature, which are nearly contemporary with our work. They are the first to focus on UDA's adversarial

robustness, but we would like to point out the clear differences from our work. RFA leverages external adversarially pre-trained ImageNet [35] models as teacher models to distill robustness knowledge. Its performance is highly sensitive to the teacher models' setup, such as perturbation budget, architecture and the number of teachers. AT on ImageNet is very expensive, so it is not always easy to obtain the preferred teacher models. In contrast, we propose a method that directly performs AT on a given UDA task, enjoying maximum flexibility. ASSUDA aims at semantic segmentation and considers only weak black-box attacks. It employs an external pre-trained UDA model to produce pseudo labels for target data, then uses the pseudo labels to do AT. However, we show that this approach has suboptimal accuracy and robustness against white-box attacks. In contrast, our method is robust under both black-box and white-box settings.

10.2 Exploring Adversarial Training for Unsupervised Domain Adaptation

This section systematically studies multiple variants of AT to explore suitable AT methods for UDA. Then we finalize the proposed ARTUDA accordingly. Here we conduct a set of experiments on the VisDA-2017 [167] dataset. We employ DANN [53] as the UDA algorithm with ResNet-50 [69] backbone. The white-box FGSM [60] attack with a perturbation budget of $\epsilon = 3$ is used for both AT and testing. Following the practice of [6, 245], we assume that attackers have the labels of the target dataset to generate adversarial examples. The rationale behind these settings is that (i) most existing UDA approaches [146, 210] are based on DANN's key idea, so DANN is a fair representative; (ii) the white-box threat model is the strongest attack setting, which

has been considered a standard evaluation protocol for defenses [5, 150, 240, 256]. In the following, we continue using the related notations and equations described in Chapter 2.

Preliminary. Given a labeled source dataset $\mathbb{D}_s = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ and an unlabeled target dataset $\mathbb{D}_t = \{x_t^i\}_{i=1}^{n_t}$ with n_s and n_t number of samples, respectively, a typical UDA model learns a feature extractor F and a classifier C on top of F. Given an input image x, we express its feature space representation as $F(x)$ and its output logits as $C(x)$, where we use $C(x)$ as a simplification of the formal expression $C(F(x))$. The objective function of an UDA model can be written as:

$$\mathcal{L}_{CE}(C(x_s), y_s) + \mathcal{L}_{DA}(x_s, x_t), \tag{10.1}$$

where $\mathcal{L}_{CE}$ is the standard cross-entropy loss, and $\mathcal{L}_{DA}$ is the domain adaptation loss defined by each UDA approach. One of the most common $\mathcal{L}_{DA}$ is the adversarial loss introduced by DANN [53], which is defined as:

$$\mathcal{L}_{DA}(x_s, x_t) = \mathbb{E}[logD(F(x_s))] + \mathbb{E}[1 - (logD(F(x_t)))], \tag{10.2}$$

where D is a domain discriminator used to encourage domain-invariant features.

AT is formulated as:

$$\min_{F,C} \mathbb{E}\left[\max_{\delta \in \mathbb{S}} \mathcal{L}(C(\tilde{x}), y)\right], \tag{10.3}$$

where $\tilde{x} = x + \delta$ is the generated adversarial example bounded by a perturbation set $\mathbb{S}$. For image classification tasks, $\mathcal{L}$ is the cross-entropy loss $\mathcal{L}_{CE}$. PGD [150], the most widely-used attack generates $\tilde{x}$ by:

$$x^{j+1} = Proj_{x,\epsilon}^{L_p}\{x^j + \alpha \cdot sign(\bigtriangledown_{x^j} \mathcal{L}(C(x^j), y))\}; \tag{10.4}$$

$\tilde{x} = x^{j_{max}}$, where j_{max} is the maximum number of attack iterations and ϵ denotes an L_p-norm perturbation budget.

10.2.1 Conventional Adversarial Training on Unsupervised Domain Adaptation

We start with applying a conventional AT [150] to DANN to see its effectiveness. That is, the AT is performed on only the labeled source data, i.e., apply Eq. (10.3) on source dataset $\mathbb{D}_s$. Therefore, the objective of the DANN model becomes:

$$\mathcal{L}_{CE}(C(\tilde{x}_s), y_s) + \mathcal{L}_{DA}(\tilde{x}_s, x_t). \tag{10.5}$$

It is reasonable to expect that Conventional AT cannot fully benefit target domain robustness, as source domain robustness may not perfectly transfer to the target domain due to domain shift. As reported in Table 10.1, compared to the Natural Training baseline (i.e., train with only clean data), Conventional AT indeed improves robustness to a certain extent but is not significant. Also, the clean accuracy is largely decreased. Hence, we argue that applying AT directly on the target data is important.

A naive way of applying AT on the target data is to produce pseudo labels y'_t using an external pre-trained UDA model. ASSUDA [245] resorts to this idea and applies it to the UDA semantic segmentation problem. Note that ASSUDA only evaluates black-box robustness. Here we implement the *Pseudo Labeling* idea on image classification and observe its white-box robustness. We use a naturally trained DANN as the pseudo-labeler. The objective of Pseudo Labeling approach is as follows:

$$\mathcal{L}_{CE}(C(x_s), y_s) + \mathcal{L}_{CE}(C(\tilde{x}_t), y'_t) + \mathcal{L}_{DA}(x_s, \tilde{x}_t). \tag{10.6}$$

Table 10.1: Results (%) of Conventional AT and our Self-Supervised AT on the VisDA-2017 dataset.

Training method	Clean	FGSM
Natural Training	73.2	21.2
Conventional AT [150]	62.9 (-10.3)	27.1 (+5.9)
Pseudo Labeling	33.1 (-40.1)	27.1 (+5.9)
Self-Supervised AT-L1	56.2 (-17.0)	15.8 (-5.4)
Self-Supervised AT-L2	51.3 (-21.9)	26.0 (+4.8)
Self-Supervised AT-KL	67.1 **(-6.1)**	**35.0 (+13.8)**

In Table 10.1, we find that Pseudo Labeling's robustness is not better than Conventional AT, and the clean accuracy drops dramatically. We believe that the label noise problem is inevitable in pseudo labels y'_t and limits model performance. This motivates us to explore a new AT method that can be directly performed on the target domain.

10.2.2 Self-Supervised Adversarial Training

Inspired by [97, 256], we seek to use clean target data's logits $C(x_t)$ as a self-supervision signal to generate adversarial examples $\tilde{x}_t$. Based on the min-max optimization for AT [150], we generate $\tilde{x}_t$ by maximizing the difference between $C(x_t)$ and $C(\tilde{x}_t)$, and minimize that difference to train a model. With this idea, we can generate adversarial examples via self-supervision and perform a kind of AT for the target domain. We call it *Self-Supervised Adversarial Training* or *Self-Supervised AT*. In other words, to generate $\tilde{x}_t$, Self-Supervised AT changes the FGSM formulation to:

$$x_t^{j+1} = \Pi_{\|\delta\|_p \leq \epsilon} \left(x_t^j + \alpha \cdot sign(\bigtriangledown_{x_t^j} \mathcal{L}(C(x_t^j), C(x_t))) \right), \tag{10.7}$$

and $\tilde{x}_t = x_t^{j_{max}}$. To adversarially train an UDA model, Self-Supervised AT changes the PGD-AT [150] formulation to:

$$\min_{F,C} \mathbb{E}\left[\max_{\|\delta\|_p \leq \epsilon} \mathcal{L}\big(C(\tilde{x}_t), C(x_t)\big)\right]. \tag{10.8}$$

$\mathcal{L}$ is a loss function that encourages the logits to be similar. Possible choices include L1 loss, L2 loss, Kullback-Leibler (KL) divergence loss, etc. Taking KL divergence loss as an example, the objective of Self-Supervised AT for UDA can be written as follows:

$$\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) + \mathcal{L}_{DA}\big(x_s, \tilde{x}_t\big), \tag{10.9}$$

where $[\cdot]_{sg}$ denotes the stop-gradient operator [215] constraining its operand to be a non-updated constant. We do not expect that Self-Supervised AT is as robust as conventional supervised AT since the ground-truth labels y are always the strongest supervision. However, given that target domain labels y_t are unavailable, we believe that the clean logits $C(x_t)$ could be a good self-supervision signal.

Table 10.1 shows that Self-Supervised AT-L1 and Self-Supervised AT-L2 are not effective, while Self-Supervised AT-KL achieves excellent results. Self-Supervised AT-KL increases robust accuracy over Natural Training by 13.8%, which is much better than Conventional AT. It also maintains decent clean accuracy. These results demonstrate that our Self-Supervised AT strategy is effective, but the choice of the loss function is critical, where KL divergence loss is the preferred one.

10.2.3 On the Effects of Clean and Adversarial Examples in Self-Supervised Adversarial Training.

Let us revisit the results of the last experiment from another perspective. We observe a trade-off between clean performance and robustness, and the upper part of Table 10.2 illustrates this point more clearly. Specifically, from Natural Training and Conventional AT, we can see that replacing clean images x_s by adversarial examples $\tilde{x}_s$ increases robust accuracy but decreases clean accuracy. A similar trade-off can be found between Natural Training and Self-Supervised AT-KL, which train with x_t and $\tilde{x}_t$, respectively. This interests us to further investigate the usage of the four data types $\{x_s, \tilde{x}_s, x_t, \tilde{x}_t\}$ in the AT for UDA. Self-Supervised AT-KL outperforms Conventional AT in terms of both clean and robust accuracies, indicating that using $\tilde{x}_t$ is more efficient than $\tilde{x}_s$, so we start with Self-Supervised AT-KL as a baseline.

First, we add x_t to Self-Supervised AT-KL. This turn out SSAT-s-t-t̃-1 and SSAT-s-t-t̃-2, where SSAT-s-t-t̃-1's domain adaptation loss is $\mathcal{L}_{DA}(x_s, x_t)$, while SSAT-s-t-t̃-2 involves another term and becomes $\mathcal{L}_{DA}(x_s, x_t) + \mathcal{L}_{DA}(x_s, \tilde{x}_t)$. In other words, SSAT-s-t-t̃-1 explicitly transfers the supervised knowledge from x_s to only x_t, while SSAT-s-t-t̃-2 transfers to both x_t and $\tilde{x}_t$. We expect that SSAT-s-t-t̃-1 and SSAT-s-t-t̃-2 enjoy higher clean accuracy than Self-Supervised AT-KL because they involve x_t.

The lower part of Table 10.2 reports the results. We find that SSAT-s-t-t̃-1's robust accuracy drops significantly, but the clean accuracy does not improve much. In contrast, SSAT-s-t-t̃-2 largely increases both clean and robust accuracies by 5.9% and 4.4%, respectively. The improvement of clean performance matches our expectations, but we are surprised at that of robustness. We see this is due to our Self-Supervised AT's specific property. Self-Supervised AT leverages the objective

Table 10.2: Results (%) of SS-AT variants on VisDA-2017. (x_s, x_t) denotes $\mathcal{L}_{DA}(x_s, x_t)$. •: selected. —: not applicable.

Training method	x_s	$\tilde{x}_s$	x_t	$\tilde{x}_t$	(x_s, x_t)	$(x_s, \tilde{x}_t)$	$(\tilde{x}_s, x_t)$	$(\tilde{x}_s, \tilde{x}_t)$	Clean	FGSM
Natural Training	•		•		•	—	—	—	73.2	21.2
Conventional AT [150]		•	•		—	—	•	—	62.9	27.1
SS-AT-KL	•			•	—	•	—	—	67.1	35.0
SS-AT-s-t-t̃-1	•		•	•	•		—	—	67.3	27.5
SS-AT-s-t-t̃-2	•		•	•	•	•	—	—	73.0	39.4
SS-AT-s-s̃-t-t̃-1	•	•	•	•	•			•	63.4	41.6
SS-AT-s-s̃-t-t̃-2	•	•	•	•		•	•		62.8	42.3
SS-AT-s-s̃-t-t̃-3	•	•	•	•	•	•	•	•	61.3	41.6

$\mathcal{L}_{KL}\big(C(\tilde{x}_t), C(x_t)\big)$ to do AT, so $C(x_t)$'s quality is critical. Given that the labels y_t is unavailable, $\mathcal{L}_{DA}(x_s, x_t)$ can transfer the supervised knowledge to x_t and thus enhance $C(x_t)$'s quality. Therefore, adding x_t to Self-Supervised AT benefits robustness as well. This observation is different from the conventional supervised AT that exists the trade-off between performance and robustness [209, 240, 256]. We conclude that involving x_t in training does help, but an explicit supervised knowledge transfer to $\tilde{x}_t$ is needed. This is rational since $\tilde{x}_t$ plays the most important role in Self-Supervised AT, giving firm guidance to it is essential.

Second, we look into the effects of $\tilde{x}_s$ in Self-Supervised AT. We add $\tilde{x}_s$ and study three variants: SSAT-s-s̃-t-t̃-1, SSAT-s-s̃-t-t̃-2 and SSAT-s-s̃-t-t̃-3. Their differences are in their domain adaptation loss, which is also illustrated in Table 10.2. Intuitively, we expect that adding $\tilde{x}_s$ falls into the trade-off that leads to lower clean performance but better robustness, as $\tilde{x}_s$ is the conventional supervised adversarial example.

As shown in Table 10.2, all the three variants obtain lower clean accuracy and higher robust accuracy than SSAT-s-t-t̃-1 and SSAT-s-t-t̃-2, which matches our assumption. The results among these three are very close. Compared to SSAT-s-t-t̃-2, their clean accuracy drops 9.6%-11.7%, but robust accuracy only improves 2.2%-2.9%.

This is consistent with Conventional AT's result, i.e., source domain robustness is not easy to transfer to the target domain. Because training without $\tilde{x}_s$ achieves a better trade-off between performance and robustness, we use SSAT-s-t-$\tilde{t}$-2 as a baseline for the next investigation. To present our experiments more clear, in the following, we summarize the objective functions of each Self-Supervised AT variant discussed in this part:

– SSAT-s-t-$\tilde{t}$-1:

$$\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) + \mathcal{L}_{DA}\big(x_s, x_t\big). \tag{10.10}$$

– SSAT-s-t-$\tilde{t}$-2:

$$\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) + \mathcal{L}_{DA}\big(x_s, x_t\big) + \mathcal{L}_{DA}\big(x_s, \tilde{x}_t\big). \tag{10.11}$$

– SSAT-s-$\tilde{s}$-t-$\tilde{t}$-1:

$$\begin{aligned} &\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) \\ &+ \mathcal{L}_{CE}\big(C(\tilde{x}_s), y_s\big) + \mathcal{L}_{DA}\big(x_s, x_t\big) + \mathcal{L}_{DA}\big(\tilde{x}_s, \tilde{x}_t\big). \end{aligned} \tag{10.12}$$

– SSAT-s-$\tilde{s}$-t-$\tilde{t}$-2:

$$\begin{aligned} &\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) \\ &+ \mathcal{L}_{CE}\big(C(\tilde{x}_s), y_s\big) + \mathcal{L}_{DA}\big(x_s, \tilde{x}_t\big) + \mathcal{L}_{DA}\big(\tilde{x}_s, x_t\big). \end{aligned} \tag{10.13}$$

– SSAT-s-s-'t-$\tilde{t}$-3:

$$\begin{aligned} &\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) + \mathcal{L}_{CE}\big(C(\tilde{x}_s), y_s\big) \\ &+ \mathcal{L}_{DA}\big(x_s, x_t\big) + \mathcal{L}_{DA}\big(x_s, \tilde{x}_t\big) + \mathcal{L}_{DA}\big(\tilde{x}_s, x_t\big) + \mathcal{L}_{DA}\big(\tilde{x}_s, \tilde{x}_t\big). \end{aligned} \tag{10.14}$$

10.2.4 On the Effects of Batch Normalization in Self-Supervised Adversarial Training

It has been well-known that the statistic estimation of BN [82] plays an important role in both the UDA (Chapter 8 and [22, 124]) and the adversarial machine learning (Chapter 6 and [240, 236]) fields. It is worth investigating the effects of BN given these two research fields meet together in this chapter.

Recall that during training, BN computes the mean and variance of the feature space for each mini-batch, referred to as *batch statistics* [240]. Each mini-batch is normalized by its batch statistics at training time. Hence, the composition of a mini-batch defines its batch statistics, thereby affecting the normalized values of each data point's features. To observe the effects on Self-Supervised AT, we create four variants of SSAT-s-t-t̃-2. They involve the same data types $\{x_s, x_t, \tilde{x}_t\}$ into training but with different mini-batch compositions. Specifically, at each training step, Batch-st-t̃ has two mini-batches, $[x_s, x_t]$ and $[\tilde{x}_t]$; Batch-s-tt̃ has two mini-batches, $[x_s]$ and $[x_t, \tilde{x}_t]$; Batch-s-t-t̃ has three mini-batches, $[x_s]$, $[x_t]$ and $[\tilde{x}_t]$; and Batch-stt̃ has one mini-batch, $[x_s, x_t, \tilde{x}_t]$. Batch-st-t̃ is the original SSAT-s-t-t̃-2, which follows the setting of [92]. We expect that their batch statistics differences would cause different results.

Table 10.3 shows the results. As can be seen, Batch-st-t̃ achieves the highest clean accuracy, while Batch-stt̃ achieves the highest robust accuracy. We argue that in Batch-stt̃, x_s is with the same mini-batch as x_t and $\tilde{x}_t$, so it can also transfer the supervised knowledge through batch statistics. In other words, the batch statistics used to normalize x_t and $\tilde{x}_t$ contain x_s's information. This shares a similar spirit with the domain adaptation loss $\mathcal{L}_{DA}(x_s, x_t) + \mathcal{L}_{DA}(x_s, \tilde{x}_t)$ discussed in Sec. 10.2.3,

Table 10.3: Results (%) of different mini-batch combinations on the VisDA-2017 dataset.

Method	Mini-batches	Clean	FGSM
Batch-st-$\tilde{\text{t}}$	$[x_s, x_t]$, $[\tilde{x}_t]$	73.0	39.4
Batch-s-t$\tilde{\text{t}}$	$[x_s]$, $[x_t, \tilde{x}_t]$	68.2	37.0
Batch-s-t-$\tilde{\text{t}}$	$[x_s]$, $[x_t]$, $[\tilde{x}_t]$	68.2	35.5
Batch-st$\tilde{\text{t}}$	$[x_s, x_t, \tilde{x}_t]$	69.0	41.4

and we have known that it can improve robustness. For Batch-st-$\tilde{\text{t}}$, we see its high performance is due to the separation of x_t and $\tilde{x}_t$. Recall that clean and robust features have distinct characteristics [84, 209], so putting them into the same mini-batch leads to suboptimal results [240]. Batch-s-t-$\tilde{\text{t}}$, however, achieves lower performance than Batch-st-$\tilde{\text{t}}$ though it has that separation as well. The reason is that in Batch-st-$\tilde{\text{t}}$, x_s and x_t are with the same mini-batch. This encourages the knowledge transfer from x_s to x_t, similar to the spirit of the domain adaptation loss $\mathcal{L}_{DA}\left(x_s, x_t\right)$.

Both Batch-st-$\tilde{\text{t}}$ and Batch-st$\tilde{\text{t}}$ achieve a good trade-off between performance and robustness. We can choose according to the downstream application's focus.

10.2.5 Recap

In this section, we explore four main aspects of AT for UDA, including Conventional AT, our Self-Supervised AT, the effects of clean and adversarial examples in Self-Supervised AT, and the effects of BN statistics. We progressively derive the best method from each investigation, then we take Batch-st$\tilde{\text{t}}$ as our final method, referred to as ARTUDA. ARTUDA's training objective is Eq.(10.11), and Figure 10.1 offers a visualized illustration. Note that some of the other variants also have their advantages, e.g., Batch-st-$\tilde{\text{t}}$, so they are still useful for certain focusses.

10.3 Experiments

We extensively evaluate the proposed ARTUDA on five adversarial attacks, three datasets and three different UDA algorithms. We further compare ARTUDA with the nearly contemporary work, RFA [6]. An analysis of feature space is also presented.

10.3.1 Experimental Setup

Datasets. We use three UDA datasets for evaluation: VisDA-2017 [167], Office-31 [182] and Office-Home [217]. VisDA-2017 contains two domains: Synthetic and Real. There are 152,409 Synthetic and 55,400 Real images from 12 object categories in this large-scale dataset. Office-31 has three domains with 31 object categories. These are Amazon (A) with 2,817 images, Webcam (W) with 795 images, and DSLR (D) with 498 images. We employ the D $\rightarrow$ W task for our experiments. Office-Home includes four domains with 65 categories: Art (Ar) with 2,427 images, Clipart (Cl) with 4,365, Product (Pr) with 4,439 images, and Real-World (Rw) with 4,375 images. We employ the Ar $\rightarrow$ Cl task for our experiment.

Attack setting. We test UDA models' adversarial robustness against four white-box attacks, including FGSM [60], PGD [150], MI-FGSM [38] and MultAdv (proposed in Chapter 3), where PGD is the default attack unless stated otherwise. A black-box attack [164] is also considered. For AT, we use the PGD attack with $j_{max} = 3$ and $\epsilon = 3$ of L_∞-norm. If not otherwise specified, we set the same for all the attacks at testing time except that FGSM's j_{max} is 1.

Benchmark UDA algorithms. We apply ARTUDA to three common UDA algorithms, including DANN [53], JAN [147] and CDAN [146]. We use ResNet-50 [69] as a backbone for all of them. If not otherwise specified, DANN is the default UDA

algorithm in our experiments.

Baseline defenses. We employ two commonly-used conventional AT algorithms, PGD-AT [150] and TRADES [256], to be our baseline defenses. To the best of our knowledge, RFA [6] might be the only approach aiming at the same problem as ours, and we also compare with it.

Implementation details. Our implementation is based on PyTorch [166]. We adopt Transfer-Learning-library [92] to set up UDA's experimental environment and follow the training hyper-parameters used in [92]. We also use the widely-used library, AdverTorch [36], to perform adversarial attacks.

10.3.2 Evaluation Results

White-box robustness. The robustness of multiple training methods against various white-box attacks is reported in Table 10.4. Without a defense, Natural Training's accuracy drops to almost 0% under strong iterative attacks. PGD-AT and TRADES improve adversarial robustness though they are originally designed for the traditional classification task. However, they also reduce clean accuracy. The proposed method, ARTUDA, significantly increases robust accuracy. Specifically, on VisDA-2017, it achieves more than 10% and 20% higher robustness than TRADES and PGD-AT, respectively. On Office-31, its robust accuracy is higher than PGD-AT and TRADES by 25%-48% under white-box iterative attacks. On Office-Home, although TRADES is slightly more robust to white-box iterative attacks, ARTUDA has higher accuracy under clean data, FGSM and black-box attacks, leading by a decent margin. In general, ARTUDA is effective across all the five attacks on three datasets. ARTUDA's clean accuracy drops but is still the best among the defenses. It can greatly improve

Table 10.4: Results (%) of UDA models on multiple datasets under various adversarial attacks.

Dataset	Training method	Clean	FGSM	PGD	MI-FGSM	MultAdv	Black-box
VisDA-2017 [167]	Natural Training	73.2	21.2	0.9	0.5	0.3	58.3
	PGD-AT [150]	60.5	34.6	21.3	22.7	7.8	59.1
	TRADES [256]	64.0	42.1	29.7	31.2	16.4	62.6
	ARTUDA (Ours)	65.5	**52.5**	**44.3**	**45.0**	**27.3**	**65.1**
Office-31 D $\rightarrow$ W[182]	Natural Training	98.0	52.7	0.9	0.6	0.1	95.0
	PGD-AT [150]	95.3	91.8	68.2	66.5	31.4	95.3
	TRADES [256]	88.4	85.3	66.4	67.0	28.2	88.2
	ARTUDA (Ours)	96.5	**95.2**	**92.5**	**92.5**	**77.1**	**96.5**
Office-Home Ar $\rightarrow$ Cl [217]	Natural Training	54.5	26.4	4.7	2.8	2.0	53.1
	PGD-AT [150]	42.5	38.8	36.0	35.8	21.7	43.0
	TRADES [256]	49.3	45.1	**41.6**	**41.6**	**22.5**	49.4
	ARTUDA (Ours)	54.0	**49.5**	41.3	39.9	21.6	**53.9**

robustness and maintain decent clean performance simultaneously.

Black-box robustness. The robustness against black-box attacks is shown in the last column of Table 10.4. Here we consider a naturally trained DANN with ResNet-18 as a substitute model and use MI-FGSM, which has better transferability, to generate black-box adversarial examples for target models. In general, the black-box attacks hardly fool the target models. However, we find that the conventional AT approaches have lower black-box accuracy than Natural Training in some cases. This is due to their lower clean accuracy. In contrast, ARTUDA has better clean accuracy and consistently achieves the best black-box robustness across all the datasets.

Generalizability. To compare with the results of [6], in this part, we evaluate robustness against the white-box PGD attack with $j_{max} = 20$ that used in [6]. Table 10.5 reports the adversarial robustness of multiple popular UDA algorithms. All of them are vulnerable to adversarial attacks. The state-of-the-art approaches, Robust PT and RFA, show excellent effectiveness in improving robustness. We apply our ARTUDA training method to these UDA models to protect them as well. As can be seen, ARTUDA uniformly robustfies all of these models. It consistently achieves low accuracy

Table 10.5: Results (%) of UDA models on the VisDA-2017 dataset under the PGD attack. Three UDA algorithms are considered.

UDA algorithm → / Training method ↓	DANN [53]			JAN [147]			CDAN [146]		
	Clean	PGD	Drop	Clean	PGD	Drop	Clean	PGD	Drop
Natural Training	73.2	0.0	-73.2	64.2	0.0	-64.2	75.1	0.0	-75.1
PGD-AT [150]	60.5	13.3	-47.2	47.7	5.8	-41.9	58.2	11.7	-46.5
TRADES [256]	64.0	19.4	-44.6	48.7	8.5	-40.2	64.6	15.7	-48.9
Robust PT [6]	65.8	38.2	-27.6	55.1	32.2	**-22.9**	68.0	41.7	-26.3
RFA [6]	65.3	34.1	-31.2	63.0	32.8	-30.2	72.0	43.5	-28.5
ARTUDA (Ours)	65.5	**40.7**	**-24.8**	58.5	**34.4**	-24.1	68.0	**43.6**	**-24.4**

drops and the highest robust accuracy, which outperforms both Robust PT and RFA. This demonstrates that ARTUDA is generic and can be applied to multiple existing UDA algorithms.

In terms of clean data accuracy, all the defenses lose clean accuracy to a certain extent. Still, the proposed ARTUDA achieves the best or the second-best clean accuracy among these defenses. Overall, it can significantly improve robustness and maintain decent clean performance simultaneously.

10.3.3 Analysis

Stability of feature space. Small adversarial perturbations on image space are enlarged considerably in feature space [239]. Hence, the stability of the feature space can reflect a model's robustness (discussed in Chapter 5). In other words, a robust model's feature space would hardly change under an adversarial example. We compute the mean L_2-norm distance between the feature space of clean images and that of their PGD examples for our models: $\| F(x_t) - F(\tilde{x}_t) \|_2$. The features from the last conv layer of the ResNet-50 backbone are used. As can be seen in Figure 10.2, Natual Training has the largest distance, which means that its features are greatly changed

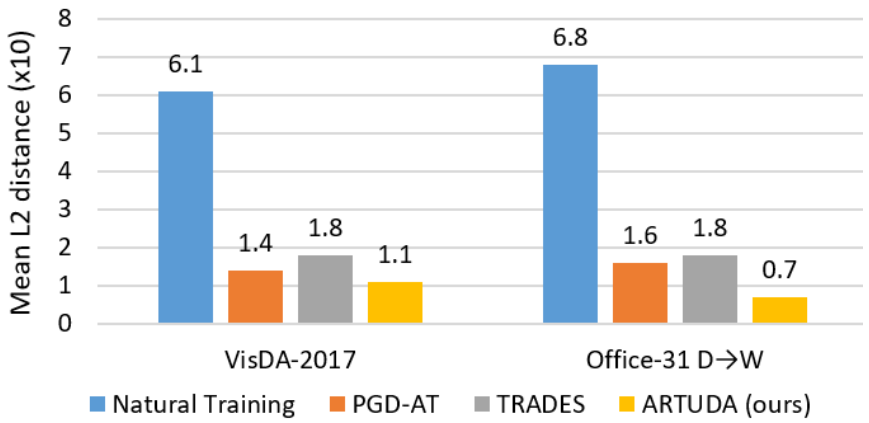

Figure 10.2: Mean L_2-norm distance between the feature space of clean images and that of their adversarial examples. The values are the mean over an entire dataset.

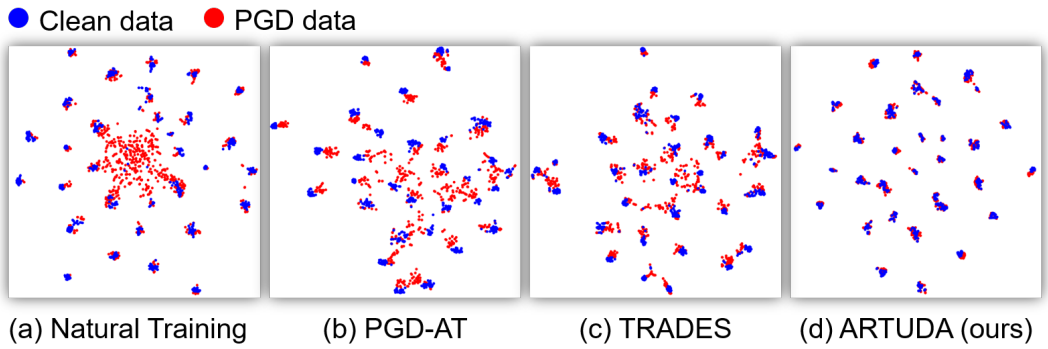

Figure 10.3: The t-SNE visualization of the feature space on the Office-31 D→W task.

when images are adversarially perturbed and thus cause wrong predictions. PGD-AT and TRADES can reduce the distance. ARTUDA attains the smallest distance on both datasets, showing that its feature space is not easily affected by adversarial perturbations.

Visualization of feature space. Figure 10.3 visualizes the different methods' feature space on the Office-31 D→W task using t-SNE [216]. The features are from the last conv layer of the ResNet-50 backbone. The PGD data in the Natural Training model are disorderly scattered and do not align with clean data. PGD-AT and TRADES

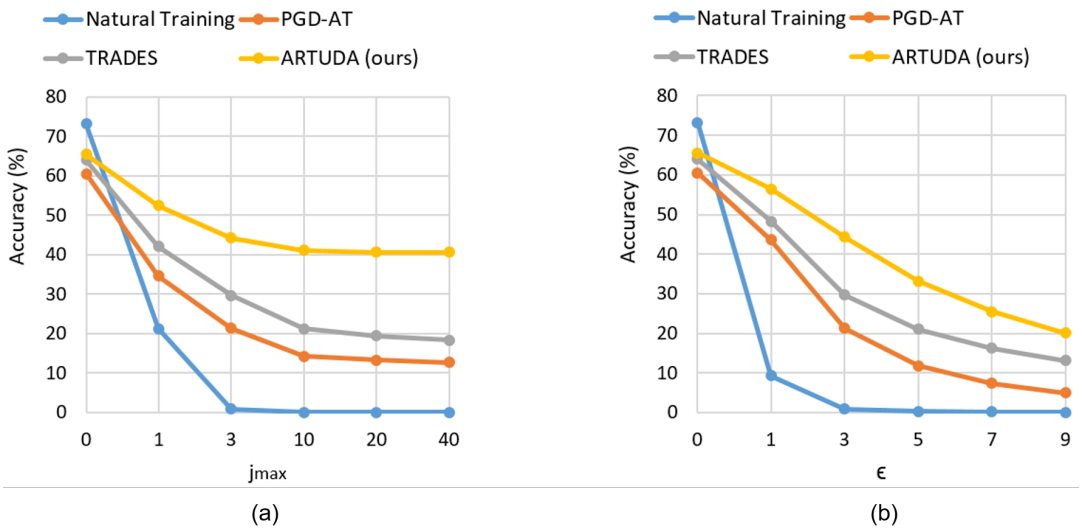

Figure 10.4: Accuracy of models under PGD attacks (a) with varied numbers of attack iterations j_{max} and (b) with varied perturbation sizes ϵ.

narrow the distribution gap to a certain extent. ARTUDA impressively aligns the feature space of PGD and clean data which almost overlap with each other. This implies that ARTUDA is effective in learning adversarially robust features. This result is consistent with the above stability analysis.

Attack budgets. We test our ARTUDA's scalability to various attack budgets. We vary the attack budgets by two aspects: the number of attack iterations j_{max} and the perturbation size ϵ. Figure 10.4 shows the results. First, we can find that the attack strength does not increase apparently along with the increase of j_{max} when $j_{max} > 3$. This observation is consistent with that of [150]. The proposed ARTUDA demonstrates stable adversarial robustness and consistently performs better than Natural Training, PGD-AT [150] and TRADES [256] under varied j_{max}. On the other hand, the attack strength dramatically increases along with the increase of ϵ. It can be seen that ARTUDA consistently shows better robustness under varied ϵ. Obviously, ARTUDA is scalable to various attack budgets.

Table 10.6: Results (%) of ARTUDA models with varied hyperparameter λ.

λ	Clean	FGSM
0.2	68.9	33.3
0.5	66.1	39.3
1.0	69.0	41.1
2.0	66.5	48.5
5.0	68.0	54.4

Loss weight of Self-Supervised AT. We can impose a hyperparameter on our ARTUDA training scheme. Specifically, we can add a loss weight λ to Eq. (10.11), and it is shown as follows:

$$\begin{aligned} &\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \lambda \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) \\ &+ \mathcal{L}_{DA}\big(x_s, x_t\big) + \mathcal{L}_{DA}\big(x_s, \tilde{x}_t\big). \end{aligned} \tag{10.15}$$

The loss weight λ controls the ratio of the Self-Supervised AT objective to the overall objective. In all of our previous experiments, we set λ to 1. In this section, we train multiple ARTUDA models with varied λ, where we use the experimental setup described in Sec. 10.2. The results are reported in Table 10.6.

We can find that the robust accuracy significantly increases along with the increase of λ, while the clean accuracy does not vary obviously. This implies that the robustness of the proposed ARTUDA can be further improved with a larger λ though it already outperforms the state-of-the-art methods.

Class-wise accuracy on VisDA-2017. In Table 10.7, we report class-wise accuracy under PGD attacks [150] on the VisDA-2017 dataset [167]. The results correspond to the PGD column in Table 10.4. We can see that ARTUDA achieves the best accuracy across the majority of the classes.

Sanity checks to evaluation. To further verify the reliability of our evaluation, we

Table 10.7: Class-wise accuracy (%) under PGD attacks on the VisDA-2017 dataset.

Training method	aero	bike	bus	car	horse	knife	motor	person	plant	skate	train	truck	Mean
Natural Training	4.8	0.9	1.5	0.0	0.2	0.9	0.3	3.2	0.2	0.1	0.7	0.0	0.9
PGD-AT [150]	49.6	20.4	15.2	8.7	34.3	7.3	27.3	32.8	35.2	**17.4**	19.8	3.2	21.3
TRADES [256]	61.8	24.5	32.0	11.4	42.9	30.6	34.1	**49.1**	50.1	5.6	33.1	4.8	29.7
ARTUDA (Ours)	**75.0**	**32.1**	**61.5**	**25.9**	**53.3**	**65.1**	**66.4**	48.2	**52.3**	9.2	**58.8**	**7.8**	**44.3**

report our results on the basic sanity checks introduced in [5]:

- Table 10.4 shows that iterative attacks (PGD and MI-FGSM) are stronger than one-step attacks (FGSM).
- Table 10.4 shows that white-box attacks are stronger than black-box attacks (by MI-FGSM).
- Unbounded attacks reach 100% attack success rate (accuracy drops to 0.0%) on all the three datasets.
- Figure 10.4 shows that increasing distortion bound increases attack success (decreases accuracy).

10.4 Summary

This chapter explores AT methods for the UDA problem. Existing AT approaches require labels to generate adversarial examples and train models, but this does not apply to the unlabeled target domain. We provide a systematic study into multiple AT variants that may suitable for UDA. This empirical contribution could offer useful insight to the research community. Based on our study, we propose ARTUDA, a novel AT method specifically designed for UDA. Our comprehensive experiments show that

ARTUDA improves robustness consistently across multiple attacks and datasets, and outperforms the state-of-the-art methods.

Chapter 11

Adaptive Batch Normalization Networks for Adversarial Robustness

11.1 Motivation

AT has been a standard foundation of modern adversarial defense approaches [150, 256]. However, it is extremely time-consuming, as it involves multi-step adversarial example generation for learning robust features. High computational cost refrains AT from being widely deployed in real-world applications. Moreover, AT is known for causing lower clean data performance [209, 256]. Image transformation-based defenses [64, 126, 244] do not use AT, but they have been proven not robust against white-box attacks [5].

In this chapter, we aim at a non-AT defense: *How to design a defense method that gets rid of AT but is still robust against strong adversarial attacks?* To answer this question, we view the adversarial robustness problem from the perspective of domain adaptation. As discussed in Chapter 6, clean data and adversarial data have distinct distributions, so we can treat adversarial examples as a kind of domain shift problem. Several studies investigate adversarial effects through domain adaptation techniques. For example, AdvProp [236] employs an auxiliary BN branch to learn separate clean and adversarial feature distributions, improving image recognition. This idea is originally from the domain adaptation field [22]. Similarly, DRRDN [247] disentangles the clean and adversarial distributions to enhance robustness, which is also motivated by domain adaptation [21]. Nevertheless, these works aim at different problem settings and still involve AT.

To our purpose, we resort to an adaptive BN idea, inspired by Test-Time Adaptation (TTA) approaches [124, 212, 223]. We propose a novel adversarial defense accordingly, referred to as Adaptive Batch Normalization Network (ABNN). ABNN employs a pre-trained and frozen substitute model to generate cleaner BN statistics,

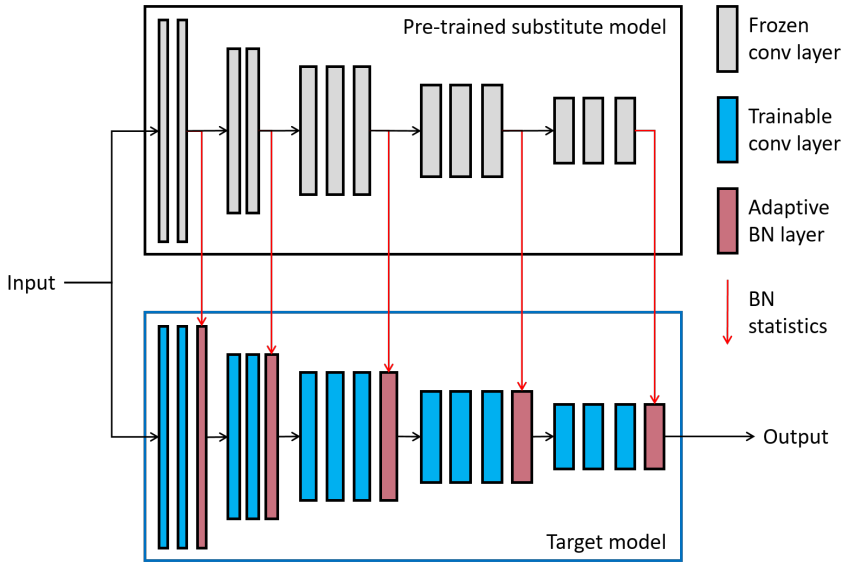

Figure 11.1: The proposed ABNN framework.

and send them to the target model. The target model is exclusively trained on clean data and learns to align the substitute model's BN statistics. Experiments show that ABNN can improve adversarial robustness and achieve higher clean data performance than AT approaches [150].

11.2 Proposed Method

As demonstrated in Chapter 6 and [236, 240], adversarial examples have different BN statistics from clean data. Such adversarial BN statistics cause accuracy drops. We see this property from a domain shift perspective and propose ABNN based on an adaptive BN idea. The framework of the proposed ABNN method is illustrated in Figure 11.1. It consists of a target model and a pre-trained substitute model. An input would pass through both models parallelly. We deploy our adaptive BN layer after each convolution block of the target model (e.g., after the conv1, conv2, conv3, conv4

and conv5 blocks of a ResNet [69]).

11.2.1 Adaptive Batch Normalization Layer

Recall that BN [82] normalizes features to address the covariate shift problem, improving training efficiency and stability. A standard BN layer is defined as:

$$z' = \gamma \left[\frac{z - \mu(z)}{\sigma(z)} \right] + \beta, \tag{11.1}$$

where z is the input feature, z' is the normalized output feature, $\{\mu(z), \sigma(z)\}$ denotes the BN statistics of mean and standard deviation, and $\{\gamma, \beta\}$ are trainable parameters for scaling and shifting, respectively. Let us consider that given an input sample, z_t and z_s represent its features extracted by the target and substitute models, respectively. Our adaptive BN layer receives $\{\mu(z_s), \sigma(z_s)\}$, the BN statistics estimated by the substitute model, then normalizes the target model's feature z_t to z'_t by:

$$z'_t = \gamma_s \left[\sigma(z_s) \left[\frac{z_t - \mu(z_t)}{\sigma(z_t)} \right] + \mu(z_s) \right] + \beta_s. \tag{11.2}$$

Inspired by [212, 214], we train the adaptive parameters $\{\gamma_s, \beta_s\}$ via the AdaIN [79] encoding layer, i.e., $\{\gamma_s, \beta_s\} = AdaIN(z_s)$. $AdaIN$ encodes the substitute model's feature z_s to derive the adaptive parameters used to align the BN statistics of the two models.

11.2.2 Training and Inference

We first pre-train the substitute model on one or multiple large-scale datasets (e.g., ImageNet [35]), where pre-training datasets are different from the target task dataset. This pre-training stage aims to learn a good feature extractor that can extract semantically

meaningful features, thereby acquiring clean and high-quality BN statistics.

Next, we train the target model with the target task dataset. The target model learns its model parameters and the adaptive parameters $\{\gamma_s, \beta_s\}$. At this stage, the substitute model sends its corresponding BN statistics $\{\mu(z_s), \sigma(z_s)\}$ to the target model's adaptive BN layers, and the substitute model itself is frozen on training. Both the pre-training and target task training stages train on clean data exclusively without AT.

At inference time, the pipeline follows the same forward pass as the target task training stage. Under adversarial attacks, the target model's BN statistics are perturbed, resulting in indiscriminate features. In comparison, the substitute model's BN statistics are relatively unaffected even under white-box attacks, since the adversary focuses more on the target model to attack the target task. Moreover, the substitute model is pre-trained on large-scale datasets different from the target task dataset, making it harder for the adversary to transfer the attack to the substitute model. Our adaptive BN layer can adapt the substitute model's cleaner BN statistics to the target model, mitigating the adversarial effects in the target model's features.

11.2.3 Discussion

ABNN does not rely on AT, so it is much more training-efficient. We present an analysis of training complexity in Sec. 11.3.3. Besides, the pre-trained substitute model can be reused for any number of downstream target task models, saving additional training time. Avoiding AT enjoys better clean data performance as well. Furthermore, compared to image transformation-based defenses, the entire ABNN framework is fully differentiable and thus does not cause obfuscated gradients.

On the other hand, ABNN is partly related to the *adaptive test-time defense* [33], a defense category that delivers adaptive defense mechanisms at test time. However, these adaptive test-time defenses involve iterative optimization during inference, significantly increasing inference computation. In contrast, our ABNN does not have test time optimization. The only extra computation overhead is the substitute model, which is much lower than optimization. Their comparison is similar to the relation between standard TTA [223] and on-the-fly adaptation [212]. In short, the proposed method takes adversarial robustness, clean data performance, training and inference efficiency into consideration, achieving a good balance among these aspects.

11.3 Experiments

11.3.1 Experimental Setup

We evaluate our method on CIFAR-10 [103], an image classification dataset that comprises 60,000 images with size 32×32 from 10 classes. We employ a ResNet-18 [69] as the backbone network of the target model, and an ImageNet [35] pre-trained VGG-19 [195] (with BN version) as the substitute model. At training time, the substitute model is frozen, and the target model is trained by the SGD optimizer.

We use the PGD [150] attack to evaluate adversarial robustness, where we set attack strength $\epsilon = 8/255$ and the number of attack iterations $t_{max} = 5$. All the attacks are conducted under the white-box setting, i.e., we generate adversarial examples upon the entire framework, so the adversaries are fully aware of the defense. Experiments are implemented by PyTorch [166] and performed on a single NVIDIA RTX 2080 Ti GPU.

Table 11.1: Evaluation results (%) on CIFAR-10.

Method	Clean	PGD
No Defense	93.4	0.0
PGD-AT [150]	83.3	51.6
ABNN (Ours)	87.5	31.5

11.3.2 Evaluation Results

Table 11.1 reports the experimental results. We can observe that the proposed ABNN significantly improves robust accuracy from 0% to 31.5% on CIFAR-10 without using AT. On the other hand, ABNN only sacrifices 5.9% clean accuracy, while PGD-AT [150] sacrifices 10.1%. Hence, although ABNN is not as robust as PGD-AT, it enjoys higher clean accuracy.

11.3.3 Training Time Complexity

Let us set each network pass (i.e., a forward pass or a backward pass) to have N computational complexity, and let us suppose that ABNN's target network and substitute network have the same complexity. Therefore, ABNN spends $2N$ on a forward pass, for it needs to pass through both networks. Since the substitute model is frozen during training, ABNN spends N on a backward pass (passes through the target network only). The total complexity of a training step is $2N + N = 3N$.

PGD-AT requires generating multi-step adversarial examples for training. It spends $2N$ on each attack iteration (a forward pass plus a backward pass). It also spends $2N$ on training model parameters at each training step. Therefore, if the number of attack iterations is t_{max}, the complexity of a single complete training step would be $(t_{max} + 1) \cdot 2N$. Hence, PGD-AT has $(t_{max} + 1) \cdot 2N/3N \simeq 0.67(t_{max} + 1)$ times

more training complexity than ABNN, which linearly increases along with the AT's t_{max}. It clearly demonstrates that ABNN is much more efficient in terms of training computation.

11.4 Summary

In this chapter, we propose a non-AT adversarial defense method, namely ABNN. With cleaner BN statistics sent from a pre-trained substitute mode, it is able to mitigate adversarial effects and thus improve robustness. Moreover, because ABNN avoids AT, it is not only much more training-efficient but also achieves better clean data performance. Adversarial robustness via domain adaptation ideas is less explored. We demonstrate that this is a promising direction and worth further exploration.

Chapter 12

Conclusion

This dissertation studies the robustness of deep learning-based computer vision models. First, we robustify computer vision models against adversarial examples, where our research covers novel attacks, empirical defenses, generalizable defenses, and defenses for less explored tasks. Next, we improve the robustness against domain shifts via domain adaptation, covering both UDA and SFDA. Finally, we explore the intersection of adversarial robustness and domain adaptation, which covers adversarial defense for domain adaptation and adversarial defense via domain adaptations. These essential topics are attracting more and more attention from computer vision and artificial intelligence communities. We hope this dissertation, which aims at more robust, reliable and trustworthy computer vision, will contribute to the research community.

12.1 Future Directions

In the future, we plan to extend this thesis in several interesting directions. They comprise three aspects: exploring the benefits of learning with perturbations, investigating real-world domain adaptation for video data, and advancing toward broader applications.

12.1.1 Benefits of Learning with Perturbations

In Part I, we study the case that perturbations are carefully crafted to be "adversarial", which are malicious to DNNs. However, perturbations can also be beneficial. For instance, learning with adversarial examples in proper ways can improve image recognition performance [74, 236]. This helps DNNs learn robust and meaningful feature representations that may not be acquired from conventional data augmentation

approaches.

Moreover, perturbations can be generated as visual prompts [8, 89]. A visual prompt is a type of visual cue that helps a pre-trained vision model adapt to downstream tasks. The prompt is trained by backpropagation, then the optimized prompt is added to input data at inference time. Such prompts may not interpretable to human eyes but look like perturbations. Rather than directly adapting a large pre-trained model to downstream tasks, visual prompting reframes the downstream tasks to resemble those solved during the original model pre-training. This leads to much more parameter-efficient tuning without requiring fine-tuning the model parameters. The generation and visual form of visual prompts are highly similar to adversarial perturbations, which is another showcase that perturbations can be beneficial.

Learning robust representations and visual prompting are emerging topics. Their potential and applications are worth further exploring.

12.1.2 Real-World Domain Adaptation for Video Data

In Part II, we study two domain adaptation scenarios, UDA and SFDA. In addition to UDA and SFDA, we would face more adaptation constraints in the real world. For example, Test-Time Adaptation (TTA) [223] is of broad interest to the research community as well. In many practical applications, a model needs to adapt itself to new data domains at test time. Compared to SFDA having unlabeled target training data, only unlabeled target test data are available for the TTA setting. TTA enables models to be more flexible when the target domains continuously change. Recently, on-the-fly adaptation [212] is introduced, which can adapt the model to a new test sample on-the-fly without updating the model parameters. This avoids backpropagation during

testing and thus further increases model flexibility.

Similar to SFDA, TTA, on-the-fly adaptation, and many other real-world adaptation settings (e.g., continual domain adaptation [219], versatile domain adaptation [94]) are less explored for video data. Chapter 9 has shown that video-based methods can significantly improve performance on video data. Hence, developing video-based methods for real-world adaptation is a promising topic. We plan to explore more advanced self-supervised learning techniques, such as Barlow Twins [254] and Masked Autoencoders [66], for our purposes.

12.1.3 Toward Broader Applications

This thesis focuses on computer vision tasks. We plan to bring this thesis's ideas to broader applications, e.g., speech processing, Natural Language Processing (NLP), and medical image analysis. Specifically, we can explore the connections between existing adversarial defense methods of image recognition (Part I) and speech recognition [96], then develop a robust multi-modal framework. Furthermore, the robust one-class algorithm proposed in Chapter 7 may also be extended to the one-class learning problems of speech [257]. For NLP, generative text detection [181] will be important for the ethical and reliable use of large language models [16]. For medical image analysis, we can leverage adversarial examples to do data augmentation, improving medical image segmentation [163]. Besides, existing approaches focus more on Computed Tomography (CT) images than Cone Beam Computed Tomography (CBCT) images. The domain adaptation algorithms proposed in Part II may be useful for learning CT-to-CBCT knowledge transfer. Lastly, we plan to explore possible real-world deployments of this thesis's ideas, such as automated driving systems [27, 31,

80, 133, 134, 135, 136], robotics [20, 242] and healthcare [158], as long-term goals.

In summary, this thesis can be advanced to broader artificial intelligence safety and application fields. In general, we can propose robust methods, leverage ideas from robustness to improve performance, and further explore diverse applications. We also present several specific directions to go. We hope this dissertation will shed light on future research.

References

[1] Sravanti Addepalli, Vivek BS, Arya Baburaj, Gaurang Sriramanan, and R. V. Babu. "Towards achieving adversarial robustness by enforcing feature consistency across bit planes". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[2] Hiroyasu Akada, Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. "Self-supervised learning of domain invariant features for depth estimation". In: *arXiv preprint arXiv: 2106.02594*. 2021.

[3] Inigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C. Murillo. "Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[4] Amir Atapour-Abarghouei and Toby P Breckon. "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[5] Anish Athalye, Nicholas Carlini, and David Wagner. "Obfuscated gradients give a false sense of security: circumventing defenses to adversarial Examples". In: *International Conference on Machine learning (ICML)*. 2018.

[6] Muhammad Awais, Fengwei Zhou, Hang Xu, Lanqing Hong, Ping Luo, Sung-Ho Bae, and Zhenguo Li. "Adversarial robustness for unsupervised domain adaptation". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[7] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*. 2017.

[8] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. "Exploring visual prompts for adapting large-scale models". In: *arXiv preprint arXiv: 2203.17274*. 2022.

[9] Abhijit Bendale and Terrance E. Boult. "Towards open set deep networks". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[10] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. "Mvtec ad–A comprehensive real-world dataset for unsupervised anomaly detection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[11] Arjun Nitin Bhagoji. "Dimensionality reduction as a defense against evasion attacks on machine learning classifiers". In: *arXiv preprint arXiv: 1704.02654*. 2017.

[12] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. "Adabins: Depth estimation using adaptive bins". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[13] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. "Evasion attacks against machine learning at test time". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*. 2013.

[14] Charles Bouman. *Digital Halftoning*. 2020. URL: https://engineering.purdue.edu/~bouman/ece637/notes/pdf/Halftoning.pdf.

[15] Tom Brown, Dandelion Mane, Aurko Roy, Martin Abadi, and Justin Gilmer. "Adversarial patch". In: *Conference on Neural Information Processing Systems Workshop (NeurIPSW)*. 2017.

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2020.

[17] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. "A non-local algorithm for image denoising". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.

[18] Nicholas Carlini and David Wagner. "Adversarial examples are not easily detected: Bypassing ten detection methods". In: *ACM Workshop on Artificial Intelligence and Security (AISec)*. 2017.

[19] Joao Carreira and Andrew Zisserman. “Quo vadis, action recognition? A new model and the kinetics dataset”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[20] Chun-Yu Chai, Yu-Po Wu, and Shiao-Li Tsao. “Deep depth fusion for black, transparent, reflective and texture-less objects”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2020.

[21] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. “All about structure: Adapting structural information across domains for boosting semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[22] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. “Domain-specific batch normalization for unsupervised domain adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[23] Rama Chellappa. *Can We Trust AI?* JHU Press, 2022.

[24] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. “Progressive feature alignment for unsupervised domain adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[25] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*. 2017.

[26] Pin-Yu Chen and Cho-Jui Hsieh. *Adversarial Robustness for Machine Learning*. Elsevier Science & Technology, 2022.

[27] Ping-Rong Chen, Shao-Yuan Lo, Hsueh-Ming Hang, Sheng-Wei Chan, and Jing-Jhih Lin. “Efficient road lane marking detection with deep learning”. In: *IEEE International Conference on Digital Signal Processing (DSP)*. 2018.

[28] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. “A simple framework for contrastive learning of visual representations”. In: *International Conference on Machine learning (ICML)*. 2020.

[29] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. “Crdoco: Pixel-level domain transfer with cross-domain consistency”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[30] Boris Chidlovskii, Stephane Clinchant, and Gabriela Csurka. “Domain adaptation in the absence of source domain data”. In: *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 2016.

[31] Chiho Choi, Joon Hee Choi, Jiachen Li, and Srikanth Malla. “Shared cross-modal trajectory prediction for autonomous driving”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[32] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. “The cityscapes dataset for semantic urban scene understanding”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[33] Francesco Croce, Sven Gowal, Thomas Brunner, Evan Shelhamer, Matthias Hein, and Taylan Cemgil. “Evaluating the adversarial robustness of adaptive test-time defenses”. In: *International Conference on Machine Learning (ICML)*. 2022.

[34] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E. Kounavis, and Duen Horng Chau. “Shield: Fast, practical defense and vaccination for deep learning using jpeg compression”. In: *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 2018.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[36] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. “AdverTorch v0.1: An adversarial robustness toolbox based on pytorch”. In: *arXiv preprint arXiv: 1902.07623*. 2019.

[37] Jeff Donahue, Lisa Anne M. Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. “Long-term recurrent convolutional networks for visual recognition and description”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[38] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. “Boosting adversarial attacks with momentum”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[39] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. "Flownet: Learning optical flow with convolutional networks". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015.

[40] Bruce Draper. *Guaranteeing AI Robustness Against Deception (GARD)*. 2019. URL: https://www.darpa.mil/program/guaranteeing-ai-robustness-against-deception.

[41] Liang Du, Jingang Tan, Hongye Yang, Jianfeng Feng, Xiangyang Xue, Qibao Zheng, Xiaoqing Ye, and Xiaolin Zhang. "Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[42] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. "A study of the effect of jpg compression on adversarial images". In: *International Society for Bayesian Analysis World Meeting*. 2016.

[43] Glenn R. Easley, Vishal M. Patel, and Jr. Dennis M. Healy. "Inverse halftoning using a shearlet representation". In: *Wavelets XIII*. 2009.

[44] David Eigen, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2014.

[45] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. "Exploring the landscape of spatial robustness". In: *International Conference on Machine learning (ICML)*. 2019.

[46] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. "Robust physical-world attacks on deep learning visual classification". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[47] Robert W. Floyd and Laurence Steinberg. "An adaptive algorithm for spatial grey scale". In: *Proceedings of the Society of Information Display*. 1976.

[48] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. "Deep ordinal regression network for monocular depth estimation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[49] Thomas Funkhouser. *Image Quantization, Halftoning, and Dithering*. 2000. URL: https://www.cs.princeton.edu/courses/archive/fall00/cs426/lectures/dither/dither.pdf.

[50] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. “Semantic video cnns through representation warping”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[51] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. “Virtual worlds as proxy for multi-object tracking analysis”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[52] Yaroslav Ganin and Victor Lempitsky. “Unsupervised domain adaptation by backpropagation”. In: *International Conference on Machine learning (ICML)*. 2015.

[53] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. “Domain-adversarial training of neural networks”. In: *Journal of Machine Learning Research (JMLR)*. 2016.

[54] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. “Unsupervised cnn for single view depth estimation: Geometry to the rescue”. In: *European Conference on Computer Vision (ECCV)*. 2016.

[55] Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for autonomous driving? The kitti vision benchmark suite”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[56] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. “Unsupervised representation learning by predicting image rotations”. In: *International Conference on Learning Representations (ICLR)*. 2018.

[57] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. “Unsupervised monocular depth estimation with left-right consistency”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[58] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. “Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[59] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial nets”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2014.

[60] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”. In: *International Conference on Learning Representations (ICLR)*. 2015.

[61] Adam Goodge, Bryan Hooi, See Kiong Ng, and Wee Siong Ng. “Robustness of autoencoders for anomaly detection under adversarial impact”. In: *International Joint Conference on Artificial Intelligence (IJCAI)*. 2020.

[62] Joseph W. Goodman. “Speckle phenomena in optics: Theory and applications”. In: *Journal of Statistical Physics*. 2008.

[63] Dayan Guan, Jiaxing Huang, Aoran Xiao, and Shijian Lu. “Domain adaptive video segmentation via temporal consistency regularization”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[64] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. “Countering adversarial images using input transformations”. In: *International Conference on Learning Representations (ICLR)*. 2018.

[65] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. “Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?” In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[66] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. “Masked autoencoders are scalable vision learners”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[67] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. “Momentum contrast for unsupervised visual representation learning”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[68] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. “Mask r-cnn”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[70] Dan Hendrycks and Kevin Gimpel. “Gaussian error linear units (gelus)”. In: *arXiv preprint arXiv: 1606.08415*. 2016.

[71] Dan Hendrycks and Kevin Gimpel. “Early methods for detecting adversarial images”. In: *International Conference on Learning Representations Workshop (ICLRW)*. 2017.

[72] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. “Using self-supervised learning can improve model robustness and uncertainty”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2019.

[73] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. “Natural adversarial examples”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[74] Charles Herrmann, Kyle Sargent, Lu Jiang, Ramin Zabih, Huiwen Chang, Ce Liu, Dilip Krishnan, and Deqing Sun. “Pyramid adversarial training improves vit performance”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[75] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. “Cycada: Cycle-consistent adversarial domain adaptation”. In: *International Conference on Machine learning (ICML)*. 2018.

[76] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. “Temporally distributed networks for fast video semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[77] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. “Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2021.

[78] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. “Rda: Robust domain adaptation via fourier adversarial attacking”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[79] Xun Huang and Serge Belongie. “Arbitrary style transfer in real-time with adaptive instance normalization”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[80] Shang-Wei Hung, Shao-Yuan Lo, and Hsueh-Ming Hang. “Incorporating luminance, depth and color information by a fusion-based network for semantic segmentation”. In: *IEEE International Conference on Image Processing (ICIP)*. 2019.

[81] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. “Flownet 2.0: Evolution of optical flow estimation with deep networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[82] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International Conference on Machine learning (ICML)*. 2015.

[83] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. “Image-to-image translation with conditional adversarial networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[84] Takahiro Itazuri, Yoshihiro Fukuhara, Hirokatsu Kataoka, and Shigeo Morishima. “What do adversarially robust models look at?” In: *arXiv preprint arXiv: 1905.07666*. 2019.

[85] Samvit Jain, Xin Wang, and Joseph E Gonzalez. “Accel: A corrective fusion network for efficient semantic segmentation on video”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[86] Eric Jang, Shixiang Gu, and Ben Poole. “Categorical reparameterization with gumbel-softmax”. In: *International Conference on Learning Representations (ICLR)*. 2017.

[87] John F. Jarvis, Charles N. Judice, and William H. Ninke. “A survey of techniques for the display of continuous tone pictures on bilevel displays”. In: *Computer Graphics and Image Processing*. 1976.

[88] Malhar Jere, Sandro Herbig, Christine Lind, and Farinaz Koushanfar. “Principal component properties of adversarial samples”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2020.

[89] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. “Visual prompt tuning”. In: *European Conference on Computer Vision (ECCV)*. 2022.

[90] Xiaojun Jia, Xingxing Wei, and Xiaochun Cao. “Identifying and resisting adversarial videos using temporal consistency”. In: *arXiv preprint arXiv: 1909.04837*. 2019.

[91] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. “Comdefend: An efficient image compression model to defend adversarial examples”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[92] Junguang Jiang, Baixu Chen, Bo Fu, and Mingsheng Long. *Transfer-Learning-library*. 2020. URL: https://github.com/thuml/Transfer-Learning-Library.

[93] Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. "Black-box adversarial attacks on video recognition models". In: *ACM International Conference on Multimedia (ACM MM)*. 2019.

[94] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. "Minimum class confusion for versatile domain adaptation". In: *European Conference on Computer Vision (ECCV)*. 2020.

[95] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution". In: *European Conference on Computer Vision (ECCV)*. 2016.

[96] Sonal Joshi, Jesús Villalba, Piotr Żelasko, Laureano Moro-Velázquez, and Najim Dehak. "Study of pre-processing defenses against adversarial attacks on state-of-the-art speaker recognition systems". In: *IEEE Transactions on Information Forensics and Security (TIFS)*. 2021.

[97] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. "Adversarial logit pairing". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2018.

[98] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. "Supervised contrastive learning". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2020.

[99] Donghyun Kim, Yi-Hsuan Tsai, Bingbing Zhuang, Xiang Yu, Stan Sclaroff, Kate Saenko, and Manmohan Chandraker. "Learning cross-modal contrastive features for video domain adaptation". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[100] Kaleab A. Kinfu and René Vidal. "Analysis and extensions of adversarial training for video classification". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*. 2022.

[101] Diederik P. Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *International Conference on Learning Representations (ICLR)*. 2015.

[102] Diederik P. Kingma and Max Welling. "Auto-encoding variational bayes". In: *International Conference on Learning Representations (ICLR)*. 2014.

[103] Alex Krizhevsky. "Learning multiple layers of features from tiny images". In: 2009.

[104] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2012.

[105] Hildegard Kuehne, Hueihan Jhuang, Estıbaliz Garrote, Tomaso Poggio, and Thomas Serre. “Hmdb: A large video database for human motion recognition”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2011.

[106] Abhijit Kundu, Vibhav Vineet, and Vladlen Koltun. “Feature space optimization for semantic video segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[107] Jogendra Nath Kundu, Akshay Kulkarni, Amit Singh, Varun Jampani, and R. Venkatesh Babu. “Generalize then adapt: Source-free domain adaptive semantic segmentation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[108] Jogendra Nath Kundu, Akshay R Kulkarni, Suvaansh Bhambri, Deepesh Mehta, Shreyas Anand Kulkarni, Varun Jampani, and Venkatesh Babu Radhakrishnan. “Balancing discriminability and transferability for source-free domain adaptation”. In: *International Conference on Machine learning (ICML)*. 2022.

[109] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R. Venkatesh Babu. “Adadepth: Unsupervised content congruent adaptation for depth estimation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[110] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. “Adversarial machine learning at scale”. In: *International Conference on Learning Representations (ICLR)*. 2017.

[111] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. “Semi-supervised deep learning for monocular depth map prediction”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[112] Chung-Sheng Lai, Zunzhi You, Ching-Chun Huang, Yi-Hsuan Tsai, and Wei-Chen Chiu. “Colorization of depth map via disentanglement”. In: *European Conference on Computer Vision (ECCV)*. 2020.

[113] Cassidy Laidlaw and Soheil Feizi. “Functional adversarial attacks”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2019.

[114] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. “Perceptual adversarial robustness: defense against unseen threat models”. In: *International Conference on Learning Representations (ICLR)*. 2021.

[115] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. “Deeper depth prediction with fully convolutional residual networks”. In: *International Conference on 3D Vision (3DV)*. 2016.

[116] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database”. In: 2010.

[117] Yi-Lun Lee, Min-Yuan Tseng, Yu-Cheng Luo, Dung-Ru Yu, and Wei-Chen Chiu. “Learning face recognition unsupervisedly by disentanglement and self-augmentation”. In: *International Conference on Robotics and Automation (ICRA)*. 2020.

[118] Michael S. Lewicki and Terrence J. Sejnowski. “Learning overcomplete representations”. In: *Neural Computation*. 2000.

[119] Yu-Jhe Li, Ci-Siang Lin, Yan-Bo Lin, and Yu-Chiang Frank Wang. “Cross-dataset person re-identification via unsupervised pose disentanglement and adaptation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[120] Jiangyun Li, Yikai Zhao, Xingjian He, Xinxin Zhu, and Jing Liu. “Dynamic warping network for semantic video segmentation”. In: *Complexity*. 2021.

[121] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. “Model adaptation: Unsupervised domain adaptation without source data”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[122] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V Krishnamurthy, Amit K Roy-Chowdhury, and Ananthram Swami. “Stealthy adversarial perturbations against real-time video classification systems.” In: *Network and Distributed System Security Symposium (NDSS)*. 2019.

[123] Xin Li and Fuxin Li. “Adversarial examples detection in deep networks with convolutional filter statistics”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[124] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. “Revisiting batch normalization for practical domain adaptation”. In: *International Conference on Learning Representations Workshop (ICLRW)*. 2017.

[125] Jian Liang, Dapeng Hu, and Jiashi Feng. “Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation”. In: *International Conference on Machine learning (ICML)*. 2020.

[126] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. “Defense against adversarial attacks using high-level representation guided denoiser”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[127] Wei-An Lin, Chun Pong Lau, Alexander Levine, Rama Chellappa, and Soheil Feizi. “Dual manifold adversarial robustness: Defense against lp and non-lp adversarial Attacks”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2020.

[128] Jiang Liu, Alexander Levine, Chun Pong Lau, Rama Chellappa, and Soheil Feizi. “Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[129] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. “Future frame prediction for anomaly detection–a new baseline”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[130] Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. “Detach and adapt: Learning cross-domain disentangled deep representation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[131] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. “Efficient semantic video segmentation with per-frame inference”. In: *European Conference on Computer Vision (ECCV)*. 2020.

[132] Yuang Liu, Wei Zhang, and Jun Wang. “Source-free domain adaptation for semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[133] Shao-Yuan Lo. *Real-Time Semantic Segmentation Networks for Autonomous Driving*. National Chiao Tung University, 2019.

[134] Shao-Yuan Lo and Hsueh-Ming Hang. “Exploring semantic segmentation on the DCT representation”. In: *ACM Multimedia Asia (MMAsia)*. 2019.

[135] Shao-Yuan Lo, Hsueh-Ming Hang, Sheng-Wei Chan, and Jing-Jhih Lin. “Efficient dense modules of asymmetric convolution for real-time semantic segmentation”. In: *ACM Multimedia Asia (MMAsia)*. 2019.

[136] Shao-Yuan Lo, Hsueh-Ming Hang, Sheng-Wei Chan, and Jing-Jhih Lin. “Multi-class lane semantic segmentation using efficient convolutional networks”. In: *IEEE International Workshop on Multimedia Signal Processing (MMSP)*. 2019.

[137] Shao-Yuan Lo, Poojan Oza, Sumanth Chennupati, Alejandro Galindo, and Vishal M. Patel. "Spatio-temporal pixel-level contrastive learning-based source-free domain adaptation for video semantic segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[138] Shao-Yuan Lo, Poojan Oza, and Vishal M. Patel. "Adversarially robust one-class novelty detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*. 2022.

[139] Shao-Yuan Lo and Vishal M. Patel. "Defending against multiple and unforeseen adversarial videos". In: *IEEE Transactions on Image Processing (T-IP)*. 2021.

[140] Shao-Yuan Lo and Vishal M. Patel. "Error diffusion halftoning against adversarial examples". In: *IEEE International Conference on Image Processing (ICIP)*. 2021.

[141] Shao-Yuan Lo and Vishal M. Patel. "Multav: Multiplicative adversarial videos". In: *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2021.

[142] Shao-Yuan Lo and Vishal M. Patel. "Exploring adversarially robust training for unsupervised domain adaptation". In: *Asian Conference on Computer Vision (ACCV)*. 2022.

[143] Shao-Yuan Lo, Jeya Maria Jose Valanarasu, and Vishal M. Patel. "Over-complete representations against adversarial videos". In: *IEEE International Conference on Image Processing (ICIP)*. 2021.

[144] Shao-Yuan Lo, Wei Wang, Jim Thomas, Jingjing Zheng, Vishal M. Patel, and Cheng-Hao Kuo. "Learning feature decomposition for domain adaptive monocular depth estimation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022.

[145] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. "Learning transferable features with deep adaptation networks". In: *International Conference on Machine learning (ICML)*. 2015.

[146] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. "Conditional adversarial domain adaptation". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2018.

[147] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. "Deep transfer learning with joint adaptation networks". In: *International Conference on Machine learning (ICML)*. 2017.

[148] Adrian Lopez-Rodriguez and Krystian Mikolajczyk. “Desc: Domain adaptation for depth estimation via semantic consistency”. In: *British Machine Vision Conference (BMVC)*. 2020.

[149] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. “The concrete distribution: A continuous relaxation of discrete random variables”. In: *International Conference on Learning Representations (ICLR)*. 2017.

[150] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. “Towards deep learning models resistant to adversarial attacks”. In: *International Conference on Learning Representations (ICLR)*. 2018.

[151] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. “Anomaly detection in crowded scenes”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[152] Reza Mahjourian, Martin Wicke, and Anelia Angelova. “Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[153] Pratyush Maini, Eric Wong, and J. Zico Kolter. “Adversarial robustness against the union of multiple perturbation models”. In: *International Conference on Machine learning (ICML)*. 2020.

[154] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. “Adversarial autoencoders”. In: *International Conference on Learning Representations Workshop (ICLRW)*. 2016.

[155] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. “Least squares generative adversarial networks”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[156] Luke Melas-Kyriazi and Arjun K Manrai. “Pixmatch: Unsupervised domain adaptation via pixelwise consistency training”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[157] Moritz Menze and Andreas Geiger. “Object scene flow for autonomous vehicles”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[158] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T. Dudley. “Deep learning for healthcare: Review, opportunities and challenges”. In: *Briefings in Bioinformatics (BiB)*. 2018.

[159] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. “Image to image translation for domain adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[160] Vinod Nair and Geoffrey E. Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *International Conference on Machine Learning (ICML)*. 2010.

[161] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv: 1807.03748*. 2018.

[162] Poojan Oza and Vishal M. Patel. “Utilizing patch-level category activation patterns for multiple class novelty detection”. In: *European Conference on Computer Vision (ECCV)*. 2020.

[163] Shaoyan Pan, Shao-Yuan Lo, Min Huang, Chaoqiong Ma, Jacob Wynne, Tonghe Wang, Tian Liu, and Xiaofeng Yang. “Deep learning-based multi-organ ct segmentation with adversarial data augmentation”. In: *SPIE Medical Imaging (SPIE MI)*. 2023.

[164] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. “Practical black-box attacks against machine learning”. In: *ACM Asia Conference on Computer and Communications Security (ASIACCS)*. 2017.

[165] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. “Learning memory-guided normality for anomaly detection”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[166] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2019.

[167] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. “Visda: The visual domain adaptation challenge”. In: *arXiv preprint arXiv: 1710.06924*. 2017.

[168] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. “Ocgan: One-class novelty detection using gans with constrained latent representations”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[169] Pramuditha Perera and Vishal M. Patel. “Deep transfer learning for multiple class novelty detection”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[170] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. “Generative probabilistic novelty detection with adversarial autoencoders”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2018.

[171] Koutilya PNVR, Hao Zhou, and David Jacobs. “Sharingan: Combining synthetic and real data for unsupervised geometry estimation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[172] Roi Pony, Itay Naeh, and Shie Mannor. “Over-the-air adversarial flickering attacks against video recognition networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[173] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. “Barrage of random transforms for adversarially robust defense”. In: *IEEE/ CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[174] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. “Certified Defenses against Adversarial Examples”. In: *International Conference on Learning Representations (ICLR)*. 2018.

[175] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J. Black. “Attacking optical flow”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[176] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. “Playing for benchmarks”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[177] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2015.

[178] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[179] David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. “Incremental learning for robust visual tracking”. In: *International Journal of Computer Vision (IJCV)*. 2008.

[180] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. “Adversarially learned one-class classifier for novelty detection”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[181] Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. “Can ai-generated text be reliably detected?” In: *arXiv preprint arXiv: 2303.11156*. 2023.

[182] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. “Adapting visual category models to new domains”. In: *European Conference on Computer Vision (ECCV)*. 2010.

[183] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. “Asymmetric tri-training for unsupervised domain adaptation”. In: *International Conference on Machine learning (ICML)*. 2017.

[184] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. “Semantic foggy scene understanding with synthetic data”. In: *International Journal of Computer Vision (IJCV)*. 2018.

[185] Mayu Sakurada and Takehisa Yairi. “Anomaly detection using autoencoders with nonlinear dimensionality reduction”. In: *ACM Conference on Machine Learning for Sensory Data Analysis Workshop*. 2014.

[186] Mohammadreza Salehi, Atrin Arya, Barbod Pajoum, Mohammad Otoofi, Amirreza Shaeiri, Mohammad Hossein Rohban, and Hamid R Rabiee. “Arae: Adversarially robust training of autoencoders improves novelty detection”. In: *Neural Networks*. 2021.

[187] Mohammadreza Salehi, Ainaz Eftekhar, Niousha Sadjadi, Mohammad Hossein Rohban, and Hamid R Rabiee. “Puzzle-ae: Novelty detection in images through solving puzzles”. In: *arXiv preprint arXiv: 2008.12959*. 2020.

[188] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. “Defense-GAN: Protecting classifiers against adversarial attacks using generative models”. In: *International Conference on Learning Representations (ICLR)*. 2018.

[189] Bernhard Schölkopf, Robert C. Williamson, Alex Smola, John Shawe-Taylor, and John Platt. “Support vector method for novelty detection”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 1999.

[190] Rui Shao, Pramuditha Perera, Pong C. Yuen, and Vishal M. Patel. “Open-set adversarial defense”. In: *European Conference on Computer Vision (ECCV)*. 2020.

[191] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition". In: *ACM Conference on Computer and Communications Security (CCS)*. 2016.

[192] Yash Sharma and Pin-Yu Chen. "Attacking the madry defense model with L_1-based adversarial examples". In: *International Conference on Learning Representations Workshop (ICLRW)*. 2018.

[193] Changhao Shi, Chester Holtz, and Gal Mishne. "Online adversarial purification based on self-supervised learning". In: *International Conference on Learning Representations (ICLR)*. 2021.

[194] Inkyu Shin, Kwanyong Park, Sanghyun Woo, and In So Kweon. "Unsupervised domain adaptation for video semantic segmentation". In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2022.

[195] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *International Conference on Learning Representations (ICLR)*. 2015.

[196] Richa Singh, Mayank Vatsa, Vishal M Patel, and Nalini Ratha. *Domain adaptation for visual understanding*. Springer, 2020.

[197] Prabhu Teja Sivaprasad and Francois Fleuret. "Uncertainty reduction for model adaptation in semantic segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[198] Khurram Soomro, Amir Roshan Zamir, Mubarak Shah, Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. "Ucf101: A dataset of 101 human actions classes from videos in the wild". In: *arXiv preprint arXiv: 1212.0402*. 2012.

[199] Serban Stan and Mohammad Rostami. "Unsupervised model adaptation for continual semantic segmentation". In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2021.

[200] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks". In: *IEEE Transactions on Evolutionary Computation*. 2019.

[201] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. "Intriguing properties of neural networks". In: *International Conference on Learning Representations (ICLR)*. 2014.

[202] David M. J. Tax and Robert P. W. Duin. “Support vector data description”. In: *Machine Learning*. 2004.

[203] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. “Fooling automated surveillance cameras: Adversarial patches to attack person detection”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*. 2019.

[204] Carlo Tomasi and Roberto Manduchi. “Bilateral filtering for gray and color images”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 1998.

[205] Florian Tramèr and Dan Boneh. “Adversarial training and robustness for multiple perturbations”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2019.

[206] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. “On adaptive attacks to adversarial example defenses”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2020.

[207] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. “Ensemble adversarial training: attacks and defenses”. In: *International Conference on Learning Representations (ICLR)*. 2018.

[208] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. “Learning to adapt structured output space for semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[209] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. “Robustness may be at odds with accuracy”. In: *International Conference on Learning Representations (ICLR)*. 2019.

[210] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. “Adversarial discriminative domain adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[211] Robert Ulichney. “Digital Halftoning”. In: *MIT Press*. 1987.

[212] Jeya Maria Jose Valanarasu, Pengfei Guo, Vibashan VS, and Vishal M. Patel. “On-the-fly test-time adaptation for medical image segmentation”. In: *Medical Imaging with Deep Learning (MIDL)*. 2023.

[213] Jeya Maria Jose Valanarasu, Vishwanath A. Sindagi, Ilker Hacihaliloglu, and Vishal M. Patel. "Kiu-net: Towards accurate segmentation of biomedical images using overcomplete representations". In: *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2020.

[214] Jeya Maria Jose Valanarasu, He Zhang, Jianming Zhang, Yilin Wang, Zhe Lin, Jose Echevarria, Yinglan Ma, Zijun Wei, Kalyan Sunkavalli, and Vishal M Patel. "Interactive portrait harmonization". In: *International Conference on Learning Representations (ICLR)*. 2023.

[215] Aaron Van Den Oord, Oriol Vinyals, et al. "Neural discrete representation learning". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2017.

[216] Laurens Van der Maaten and Geoffrey E. Hinton. "Visualizing data using t-sne". In: *Journal of Machine Learning Research (JMLR)*. 2008.

[217] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. "Deep hashing network for unsupervised domain adaptation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[218] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. "Extracting and composing robust features with denoising autoencoders". In: *International Conference on Machine learning (ICML)*. 2008.

[219] Riccardo Volpi, Diane Larlus, and Grégory Rogez. "Continual adaptation of visual representations via domain randomization and meta-learning". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[220] Vibashan VS, Vikram Gupta, Poojan Oza, Vishwanath A. Sindagi, and Vishal M. Patel. "Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive Object Detection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[221] Vibashan VS, Poojan Oza, and Vishal M. Patel. "Instance relation graph guided source-free domain adaptive object detection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[222] Vibashan VS, Domenick Poster, Suya You, Shuowen Hu, and Vishal M Patel. "Meta-uda: Unsupervised domain adaptive thermal object detection using meta-learning". In: *IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022.

[223] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. “Tent: Fully test-time adaptation by entropy minimization”. In: *International Conference on Learning Representations (ICLR)*. 2021.

[224] Wenguan Wang, Tianfei Zhou, Fatih Porikli, David Crandall, and Luc Van Gool. “A survey on deep learning technique for video segmentation”. In: *arXiv preprint arXiv: 2107.01153*. 2021.

[225] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. “Exploring cross-image pixel contrast for semantic segmentation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[226] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing (T-IP)*. 2004.

[227] Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. “Transferable adversarial attacks for image and video object detection”. In: *International Joint Conference on Artificial Intelligence (IJCAI)*. 2019.

[228] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. “Sparse adversarial perturbations for videos”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2019.

[229] Eric Wong and J. Zico Kolter. “Provable defenses against adversarial examples via the convex outer adversarial polytope”. In: *International Conference on Machine learning (ICML)*. 2018.

[230] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. “Cbam: Convolutional block attention module”. In: *European Conference on Computer Vision (ECCV)*. 2018.

[231] Tong Wu, Liang Tong, and Yevgeniy Vorobeychik. “Defending against physically realizable attacks on image classification”. In: *International Conference on Learning Representations (ICLR)*. 2020.

[232] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. “Learning discriminative reconstructions for unsupervised outlier removal”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015.

[233] Chaowei Xiao, Ruizhi Deng, Bo Li, Taesung Lee, Benjamin Edwards, Jinfeng Yi, Dawn Xiaodong Song, Mingyan Liu, and Ian Molloy. “Advit: Adversarial frames identifier based on temporal consistency in videos”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[234] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. "Spatially transformed adversarial examples". In: *International Conference on Learning Representations (ICLR)*. 2018.

[235] Han Xiao, Kashif Rasul, and Roland Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms". In: *arXiv preprint arXiv: 1708.07747*. 2017.

[236] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Yuille, and Quoc V. Le. "Adversarial Examples Improve Image Recognition". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[237] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V. Le. "Smooth adversarial training". In: *arXiv preprint arXiv: 2006.14536*. 2020.

[238] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. "Mitigating adversarial effects through randomization". In: *International Conference on Learning Representations (ICLR)*. 2018.

[239] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. "Feature denoising for improving adversarial robustness". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[240] Cihang Xie and Alan Yuille. "Intriguing properties of sdversarial training at scale". In: *International Conference on Learning Representations (ICLR)*. 2020.

[241] Yun Xing, Dayan Guan, Jiaxing Huang, and Shijian Lu. "Domain Adaptive Video Segmentation via Temporal Pseudo Supervision". In: *European Conference on Computer Vision (ECCV)*. 2022.

[242] Jingxi Xu, Han Lin, Shuran Song, and Matei Ciocarlie. "Tandem3d: Active tactile exploration for 3d object recognition". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2023.

[243] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. "Learning in the frequency domain". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[244] Weilin Xu, David Evans, and Yanjun Qi. "Feature squeezing: Detecting adversarial examples in deep neural networks". In: *Network and Distributed System Security Symposium (NDSS)*. 2018.

[245] Jinyu Yang, Chunyuan Li, Weizhi An, Hehuan Ma, Yuzhi Guo, Yu Rong, Peilin Zhao, and Junzhou Huang. "Exploring robustness of unsupervised domain adaptation in semantic segmentation". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[246] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. “Generalized source-free domain adaptation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[247] Shuo Yang, Tianyu Guo, Yunhe Wang, and Chang Xu. “Adversarial robustness through disentangled representations”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2021.

[248] Xiulong Yang and Shihao Ji. “Learning with multiplicative perturbations”. In: *International Conference on Pattern Recognition (ICPR)*. 2020.

[249] Yanchao Yang and Stefano Soatto. “Fda: Fourier domain adaptation for semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[250] Rajeev Yasarla and Vishal M. Patel. “Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[251] Hao-Wei Yeh, Baoyao Yang, Pong C Yuen, and Tatsuya Harada. “Sofa: Source-data-free feature alignment for unsupervised domain adaptation”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2021.

[252] Sergey Zagoruyko and Nikos Komodakis. “Wide residual networks”. In: *British Machine Vision Conference (BMVC)*. 2016.

[253] Michał Zajac, Konrad Zołna, Negar Rostamzadeh, and Pedro O. Pinheiro. “Adversarial framing for image and video classification”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2019.

[254] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. “Barlow twins: Self-supervised learning via redundancy reduction”. In: *International Conference on Machine Learning (ICML)*. 2021.

[255] He Zhang and Vishal M. Patel. “Densely connected pyramid dehazing network”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[256] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. “Theoretically principled trade-off between robustness and accuracy”. In: *International Conference on Machine learning (ICML)*. 2019.

[257] You Zhang, Fei Jiang, and Zhiyao Duan. “One-class learning towards synthetic voice spoofing detection”. In: *IEEE Signal Processing Letters (SPL)*. 2021.

[258] Zhiwei Zhang, Shifeng Chen, and Lei Sun. “P-kdgan: Progressive knowledge distillation with gans for one-class novelty detection”. In: *International Joint Conference on Artificial Intelligence (IJCAI)*. 2020.

[259] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. “Geometry-aware symmetric domain adaptation for monocular depth estimation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[260] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. “T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks”. In: *European Conference on Computer Vision (ECCV)*. 2018.

[261] Chong Zhou and Randy C. Paffenroth. “Anomaly detection with robust deep autoencoders”. In: *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 2017.

[262] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. “Unsupervised learning of depth and ego-motion from video”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[263] Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V. K. Vijaya Kumar, and Jinsong Wang. “Confidence regularized self-training”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.