The Roles of Adversarial Examples on Trustworthiness of Deep Learning

The Roles of Adversarial Examples on Trustworthiness of Deep Learning

Tianjin Huang

The Roles of Adversarial Examples on Trustworthiness of Deep Learning
by Tianjin Huang.
Eindhoven: Technische Universiteit Eindhoven, 2023. Proefschrift.

A catalogue record is available from the Eindhoven University of Technology Library

ISBN 978-90-386-5681-6

TU/e
EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

SIKS

SIKS Dissertation Series No. 2023-07
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Keywords:	adversarial examples, adversarial robustness, optimization trajectories, nodal representation
Cover image:	Generated by DALL.E
Cover design:	Tianjin Huang
Printed by:	ipskampprinting.nl

The Roles of Adversarial Examples on Trustworthiness of Deep Learning

THESIS

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op dinsdag 28 februari 2023 om 16:00 uur

door

Tianjin Huang

geboren te Tongling, China

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

Voorzitter:	Prof. dr. M.A. (Mark) Peletier
Promotor:	Prof. dr. M. Pechenizkiy
Co-promotor:	Dr. V. Menkovski
Co-promotor:	Dr. Y. Pei
Leden:	Prof. dr. F. Roli (University of Genova)
	Prof. dr. X. Huang (University of Liverpool)
	Prof. dr. ir W.P.A.J. Michiels
	Dr. habil. C. de Campos
Adviseur:	Dr. M. Fang (University of Liverpool)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Summary

> “Or maybe it’s just that beautiful things are so easily broken by the world.”
>
> *Cassandra Clare, City of Fallen Angels*

With the widespread application of Deep learning models in domains such as finance, transportation, medicine, and security, there is a rising concern regarding the trustworthiness of these models. This concern makes it hard to deploy deep learning models in risk-sensitive tasks such as face recognition, autonomous driving, and medical diagnostic. Therefore, to improve the trustworthiness of deep learning models, several aspects should be considered such as their robustness, generalization, explainability, transparency, fairness, and privacy preservation. At the current state-of-the-art, many models are found to be vulnerable to imperceptible attacks, biased against underrepresented groups, and lacking in user privacy protection. This not only degrades the user experience but also erodes society’s trust in all artificial intelligence (AI) systems. One clear case of the model’s vulnerability is given by adversarial examples. Adversarial examples are special inputs perturbed by well-designed changes with the purpose of confusing deep learning models. It has been demonstrated that such examples can be found for a wide range of deep learning models, resulting in a great concern regarding the safety of such models. Although adversarial examples are commonly used to attack deep learning models, they play important roles in multiple aspects of DL models. Specifically, adversarial examples can be used to ❶ attack DL models, ❷ build and evaluate adversarial robust models, ❸ boost

the generalization of a model. In this work, we carry out a series of research studies delving into these three aspects of adversarial examples.

To address the first aspect, we present the Direction-Aggregation (DA) attack method, which enhances the transferability of adversarial examples and strengthens black-box attacks. DA attack smoothens the decision boundary to prevent overfitting of attack direction to the white-box model. Our experiments demonstrate that DA attack notably improves the transferability of adversarial examples.

With regards to the second aspect, we introduce Calibrated Adversarial Training (CAT), Weighted Optimization Trajectories (WOT), and Curvature-based Regularization as solutions to address robust overfitting, trade-off, and low training efficiency issues.

- CAT: CAT adapts the training data, i.e., adversarial examples, at pixel level with the goal of reducing the semantic content changes in the input. Our results show that training on the adapted inputs achieves a better trade-off in clean accuracy and robust accuracy than baselines.

- WOT: WOT refines the optimization trajectories by maximizing the robust accuracy on an unseen dataset. An intuition behind this is that a model's ability to generalize robustness to an unseen dataset is likely indicative of its ability to generalize robustness to other unseen datasets. Our results demonstrate that WOT integrates seamlessly with adversarial training methods and effectively addresses the robust overfitting issue, leading to improved adversarial robustness.

- Curvature-based Regularization. We observe that large curvatures along the Fast Gradient Signed Method (FGSM) perturbed direction result in a significant difference in the adversarial robustness performance between FGSM-based adversarial training (FGSM-AT) and Projected Gradient Descent attack based adversarial training (PGD-AT). To mitigate this, we propose combining FGSM-AT with curvature regularization to close the gap between FGSM-AT and PGD-AT. Our experiments show that this method achieves similar adversarial robustness to PGD-AT while maintaining the fast training efficiency of FGSM-AT, significantly speeding the training time of PGD-AT.

Motivated by the third aspect, we investigate the impact of adversarial examples on enhancing the generalization performance of Graph Autoencoder (GAE) and Variational Graph Autoencoder (VGAE) models. Our findings highlight two crucial factors that contribute to improved task performance: the magnitude of allowed perturbations and the strength of regularization using adversarial examples. With the optimal balance between these factors, adversarial examples can significantly improve the generalization of graph representations learned by GAE/VGAE models, resulting in better performance in node classification, link prediction, and anomaly detection tasks.

List of Publications

Conference Publications

1. **Tianjin Huang**, Tianlong Chen, Meng Fang, Vlado Menkovski, Jiaxu Zhao, Lu Yin, Yulong Pei, Decebal Constantin Mocanu, Zhangyang Wang, Mykola Pechenizkiy, and Shiwei Liu. *You Can Have Better Graph Neural Networks by Not Training Weights at All: Finding Untrained Graph Tickets*. Learning on Graphs Conference (LoG), PMLR, 2022, **Oral & Best Paper Award**.

2. **Tianjin Huang**, Vlado Menkovski, Yulong Pei, Mykola Pechenizkiy. *calibrated adversarial training*. Asian Conference on Machine Learning (ACML), PMLR, 2021.

3. **Tianjin Huang**, Vlado Menkovski, Yulong Pei, Mykola Pechenizkiy. *On Generalization of Graph Autoencoders with Adversarial Training*. Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML), Springer, 2021.

4. **Tianjin Huang**, Yulong Pei, Vlado Menkovski, Mykola Pechenizkiy. *Hop-count based self-supervised anomaly detection on attributed networks*. Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML), Springer, 2022.

5. Shiwei Liu, Tianlong Chen, Zhenyu Zhang, Xuxi Chen, **Tianjin Huang**, Ajay Kumarjaiswal, Zhangyang Wang. *Sparsity May Cry: Let Us Fail (Current) Sparse Neural Networks Together*. International Conference on Learning Representations (ICLR), PMLR, 2023.

6. Yin Lu, Vlado Menkovski, Meng Fang, **Tianjin Huang**, Yulong Pei, Mykola Pechenizkiy, Decebal Constantin Mocanu, Shiwei Liu. *Superposing Many Tickets into One: A Performance Booster for Sparse Neural Network Training*. Uncertainty in Artificial Intelligence (UAI), 2022.

7. Yin Lu, Shiwei Liu, Fang Meng, **Tianjin Huang**, Vlado Menkovski, Mykola Pechenizkiy, Decebal Constantin Mocanu, Shiwei Liu. *Lottery Pools: Winning More by interpolating Tickets without Increasing Training or Inference Cost*.Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI), 2023.

Journal Publications

8. **Tianjin Huang**, Vlado Menkovski, Yulong Pei, Yuhao Wang, Mykola Pechenizkiy. *Direction-aggregated attack for transferable adversarial examples*. ACM Journal on Emerging Technologies in Computing Systems (JETC) 18.3 (2022): 1-22.

9. Yulong Pei, **Tianjin Huang**, W. Ipenburg, Mykola Pecheniky. *ResGCN: attention-based deep residual modeling for anomaly detection on attributed networks*. Machine Learning 111.2 (2022): 519-541.

Preprint and Unpublished Manuscript

10. **Tianjin Huang**, Shiwei Liu, Tianlong Chen, Meng Fang, Li Shen, Vlado Menkovski, Lu Yin, Yulong Pei, Mykola Pechenizkiy. *In-Time Refining Optimization Trajectories Toward Improved Robust Generalization*. Submitted to Transactions on Machine Learning Research (TMLR), Under review, 2023.

11. **Tianjin Huang**, Lu Yin, Zhenyu Zhang, Li Shen, Meng Fang, Mykola Pechenizkiy, Zhangyang Wang, Shiwei Liu. *Are Large Kernels Better Teachers than Transformers for ConvNets*. Submitted to International Conference on Machine Learning (ICML), Under review, 2023.

12. **Tianjin Huang**, Vlado Menkovski, Yulong Pei, Mykola Pechenizkiy. *Bridging the Performance Gap between FGSM and PGD Adversarial Training.* arXiv preprint arxiv: 2011.05157, 2020.

13. Shiwei Liu, **Tianjin Huang**, Tianlong Chen, Zhangyang Wang. *The Counterattack of CNNs in Self-Supervised Learning: Larger Kernel Size is All You Need.*, Unpublished Manuscript, 2023.

14. Sibylle Hess, **Tianjin Huang**, Wouter Duivesteijn. *Islands of Confidence: Robust Neural Network Classification with Uncertainty Quantification*. Unpublished manuscript, 2023.

15. Yutian Liu, **Tianjin Huang**, Melvin Wong, Tao Feng, and Soora Rasouli. *RT-GCN: A Gaussian-based Spatiotemporal Graph Convolutional Network for Robust Traffic Prediction*. Submitted to Transportation Research Part C, Under review, 2023.

16. Lu Yin, Gen Li, Meng Fang, Li Shen, **Tianjin Huang**, Zhangyang Wang, Vlado Menkovski, Xiaolong Ma, Mykola Pechenizkiy, Shiwei Liu. *Dynamic Sparsity Is Channel-Level Sparsity Learner*. Submitted to International Conference on Machine Learning (ICML), Under review, 2023.

17. Jiaxu Zhao, Meng Fang, Yitong Li, **Tianjin Huang**, and Mykola Pechenizkiy. *Prefix-Debias: Using Continuous Prompts to Mitigate Biases*. Submitted to the 61st Annual Meeting of the Association for Computational Linguistics (ACL), Under review, 2023.

Publication Explanation

As the lead author of papers 1, 2, 3, 4, 8, 10, 11, and 12, I played a key role in the entire research process, including ideation, experimental design, implementation, and manuscript writing. As a co-author for papers 5, 6, 7, 9, 13, 14, 15, 16, and 17, I contributed to the ideation process and writing of papers 9, 15, and 13. I also actively participated in the experiments and provided critical feedback on papers 5, 6, 7, 14, 16, and 17.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapid advancement of deep learning (DL) models is having a significant impact on society. DL-based systems play a critical role in many aspects of our lives, from unlocking mobile phones with facial recognition to guiding autonomous vehicles, interacting with voice assistants, and recommending products online. Despite these benefits, the limitations of DL-based systems have become increasingly apparent, particularly with regard to their lack of reliability and trustworthiness. For example, safety-sensitive DL-based systems have been shown to be vulnerable to adversarial examples. As shown in [EEF$^+$18], small perturbations on the road signs make the image recognition system fail to recognize it, posing a huge threat to passenger safety. Besides, DL-based chatbot systems have been shown to be biased and unfair. Online chatbots have been observed to produce indecent and racist content [WMG17]. Furthermore, DL-based systems also pose a risk in disclosing users' private information. These vulnerabilities could make current DL-based systems unstable and cause severe disasters in the human economy and security. Among these vulnerabilities, adversarial examples pose a particularly severe challenge to DL models since they are difficult to be detected and discriminated [CW17a, SZS$^+$13] from the true examples and easy to be conducted by the attackers. Toward the goal of building trustworthy DL models, we focus on understanding and eliminating the particular yet challenging weakness of DL models– adversarial examples.

Adversarial examples are formed by adding carefully designed perturbations to the original input with the purpose of confusing deep neural networks (DNNs), leading to a wrong prediction. As shown in Figure 1.1, an input of "howler

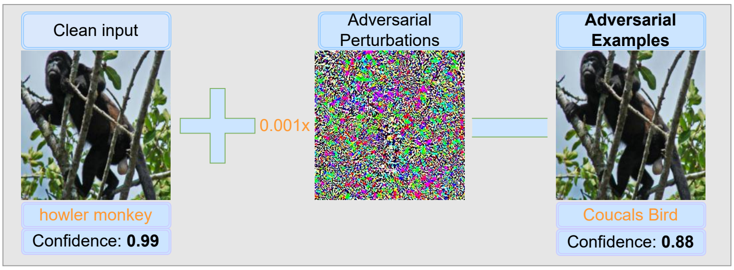


Figure 1.1: A case of adversarial examples. The prediction of the deep neural network is changed to "Coucals Bird" from "howler monkey" by adding small perturbations.

monkey" is misclassified as "Coucals Bird" with high confidence after adding extremely small perturbations.

Adversarial examples play important roles in multiple aspects of DL models. Specifically, the roles of adversarial examples on DL models can be separated into three aspects (as shown in Figure 1.2): ❶ Adversarial examples are used to attack DL models, known as adversarial attack [SZS+13, MDFFF17, MDFF16]; ❷ Adversarial examples are important tools for evaluating and boosting the model's adversarial robustness (measured by robust accuracy) [CW17b, CH20b, GSS14a, MMS+18]; ❸ Adversarial examples can potentially play a positive effect on task performance [XZZ+19]. In this thesis, we delve into each role of adversarial examples in DL models.

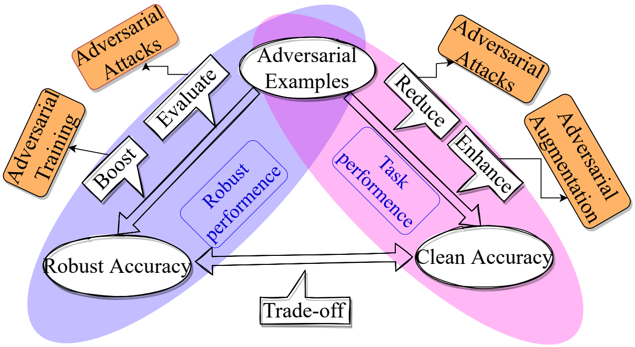


Figure 1.2: Roles of adversarial examples on DL models.

The remaining part of the Introduction to this thesis is organized as follows.

We revisit the basic definitions of adversarial examples and adversarial training in Section 1.1. Then we introduce the key research questions the thesis is focused on in Section 1.2. We explain how we studied these questions in Section 1.3. Finally, in Section 1.4, we highlight the main achievements and explain how the rest of the thesis is organized.

1.1 Background

1.1.1 Notations

We denote a C-class dataset by $D = \{(x_i, y_i)|x_i \in \mathbb{R}^d, y_i \in \mathbb{R}\}_{i=1}^n$ and a DNN function by $f_\theta : \mathbb{R}^d \to \mathbb{R}^C$. We denote cross-entropy loss by $L(\cdot)$. We denote the L_p norm by $\|\cdot\|_p$. We denote the maximum magnitude of the allowed perturbation by ϵ.

1.1.2 Adversarial examples

Adversarial examples are first discovered in [SZS+13]. It is found that small perturbations on the input can arbitrarily change the deep models' prediction. Specifically, adversarial examples can be obtained by optimizing the following objective:

$$\max_{\|\delta\|_p \le \epsilon} L(f_\theta(x+\delta), y), \tag{1.1}$$

With optimizing the objective function 1.1, the cross-entropy loss will be enlarged such that the prediction of $f_\theta(x+\delta)$ does not equal the true label y. The process of generating adversarial examples is known as adversarial attacks.

Methods for Generating Adversarial Examples. According to whether an attacker can access the target model including the model's architecture, parameters, and training data, adversarial attacks can be categorized as White-box attacks and Black-box attacks [ZL19]. White-box attacks usually generate adversarial examples by accumulating gradients with respect to maximizing the training loss. Popular white-box attacks include Limited-memory Broyden-Fletcher-Goldfarb-Shanno attack(L-BFGS) [SZS+13], Fast Gradient Sign Method (FGSM) [GSS14a], Iterative Gradient Sign Method (IGSM) [KGB16], C&W attack [CW17b], Deep Fool [MDFF16], Jacobian-based Saliency Map attack (JSMA) [PMW+16], Projected Gradient Descent attack (PGD) [MMS+17], Autoattack [CH20b] and so on. Black-box attacks can be further classified as transferability-based attacks and query-based attacks. The transferability-based black-box attack usually needs to train a surrogate model on the known trainset and then generates adversarial examples based on the surrogate model [DPSZ19, LCLS17, XZZ$^+$19, LSH$^+$20, HMP$^+$22, WZ20, HK22]. In contrast, Query-based black-box attack does not need to train a surrogate model. It generates adversarial examples based on an approximate gradient estimated by the finite difference [DCP$^+$21, IEAL18, ACFH20, BZJ$^+$20, CZS$^+$17].

Variants of Adversarial Examples. Since adversarial examples are first identified by [SZS$^+$13], various variants of adversarial examples have been explored. Su et al. [SVS19] propose one-pixel adversarial examples where only one-pixel value is changed, resulting in fooling the model's prediction. Song et al. [SEE$^+$18] and Kurakin et al. [KGB17] propose physical adversarial examples to fool the predictors by modifying visual characteristics of the real object in the physical world. For example, Athalye et al. [AEIK18] construct an adversarial object to fool the model and Eykholt et al. [EEF$^+$18] stick a sticker on the stop sign for fooling the model's detection. Xiao et al. [XZL$^+$18] propose to generate adversarial examples based on spatial transformation instead of direct manipulation of the pixel values. Moosavi et al. [MDFFF17] further demonstrate the existence of universal adversarial perturbations that cause the input to be classified.

Transferability of Adversarial Examples. An intriguing property of adversarial examples is that they can transfer among different architectures, which is first found by [LCLS17]. An explanation for this is that the decision boundary of models trained on the same train set is similar [LCLS17, HMP$^+$22]. This property is utilized to conduct the black-box attack for the unknown models. However, the transferability of adversarial examples directly generated by accumulating the gradients from the model is poor due to the adversarial examples are easily overfitting to the white-box model. Therefore, many techniques have been proposed to enhance the transferability by smoothing the decision boundaries [XZZ$^+$19, DPSZ19, LSH$^+$20, HMP$^+$22, WZ20, WWX$^+$20b, WH21, GLC20, HK22].

1.1.3 Adversarial Robustness

. Adversarial robustness refers to a model's ability to resist adversarial attacks. Formally, adversarial robustness for a model f_θ is defined as follows: the model is robust to adversarial perturbations of magnitude δ at input x if and only if [QMG$^+$19]

$$arg\max_{i\in C} f_\theta^i(x) = arg\max_{i\in C} f_\theta^i(x+\delta) \;\; \forall\delta \in \{\delta : \|\delta\|_p \le \epsilon\} \tag{1.2}$$

Since the first demonstration of the vulnerability that deep models are vulnerable to adversarial attacks [SZS+13], many methods have been proposed to improve the model's adversarial robustness including adversarial training [GSS14b, MMS+18,ZYJ+19a], input purification [LLD+18,SKC18],regularization [QMG+19, FO21, RDV18, MDFUF19]. Among these methods, many of them are proven to be ineffective in defending adversarial attacks [ACW18]. In contrast, adversarial training, i.e., training models on the on-the-fly generated adversarial examples, is the most effective method in boosting the model's adversarial robustness.

Goodfellow et al. [GSS14a] take the first attempt to train the model by including the FGSM adversarial examples, a.k.a. FGSM-adversarial training, and demonstrates its effectiveness for single-step adversarial attacks. However, FGSM-adversarial trained models are found still vulnerable to multi-step adversarial attacks. To alleviate this issue, [MMS+18] propose to train a model on adversarial examples generated by multiple steps adversarial attack (PGD-AT) and formalized it as *min-max* optimizing problem:

$$\min_{\theta} \rho^{AT}(\theta), \ \rho^{AT}(\theta) = \frac{1}{n}\sum_{i}^{n}\{\max_{\|\delta\|\leq\epsilon} L(f_{\theta}(x_i+\delta), y_i)\}, \tag{1.3}$$

where the *inner maximization* is to find the adversarial examples and ϵ is the allowed perturbation magnitude. PGD-AT effectively improves adversarial accuracy against various adversarial attacks, e.g. FGSM attack [GSS14a], PGD attack [MMS+17], AA [CH20b]. However, there are three challenges in applying PGD-AT for achieving adversarial robustness: ❶ Trade-off between Clean Accuracy and robust accuracy, ❷ Robust overfitting, ❸ Expensive training cost.

Trade-off. The trade-off between robust accuracy and clean accuracy has been widely observed [SST+18, SZC+18, ZYJ+19a, WCG+20]. This trade-off is provably shown to exist on a simple binary classification task [TSE+18a, Nak19, RXY+20]. At the same time, many techniques have been proposed to alleviate this trade-off. Alayrac et al. [AUH+19], Carmon et al. [CRS+19], Raghunathan et al. [RXY+20] show that this trade-off can be good alleviated by adding more training data. Moreover, Balaji et al. [BGH19], Zhang et al. [ZZN+20], Zhang et al. [ZXH+20a], and Huang [HMPP21] alleviate this trade-off without additional data by either adapting adversarial perturbations or re-weighting adversarial examples. Pang et al. [PLY+22] further propose to reconcile robust accuracy and clean accuracy by minimizing self-consistent robust error.

Robust Overfitting. Robust overfitting refers to the phenomenon that robust accuracy in the test set degrades severely after the first learning rate decay during the training, resulting in poor robust generalization. This phenomenon is first identified by [RWK20]. Since then, several studies have been proposed to explain and mitigate the robust overfitting issue [WXW20, SSFJ21, CZL+20, DXY+21, CZW+22, SHS21]. Chen et al. [CZL+20] show that stochastic weight average (SWA) and knowledge distillation can mitigate robust overfitting issue decently and Singla et al. [SSFJ21] found that low curvature activation helps to mitigate robust overfitting problem. Dong et al. [DXY+21] took a step further to explain that robust overfitting may be caused by the memorization of hard samples in the final phase of training. Wu et al. [WXW20], Yu et al. [YHG+21], and Stutz et al. [SHS21] demonstrate that a flattened loss landscape improves robust generalization and reduces robust overfitting problem, which is in line with the sharpness studies in standard training setting [FKMN20,JNM+19,DR17].

Expensive Training Cost. Due to the inner loop for generating adversarial examples, adversarial training usually takes more than multiple times of standard training cost, making it hard to be applied in large datasets, such as ImageNet. To alleviate this issue, Shafahi et al. [SNG+19] propose to update the model parameters and image perturbations on one simultaneous backward pass, which achieves 3-30x time faster than standard adversarial training. At the same time, Zhang et al. [ZZL+19] observe that adversarial perturbation is coupled with the first layer of the model. This observation inspires them to restrict most of the forward and backward propagation within the first layer, resulting in acceleration with 4x-5x less training time. Moreover, Zheng et al. [ZZG+20] propose to improve the training efficiency of AT by accumulating adversarial perturbations through epochs, leading to an acceleration of the training. On the other side, since FGSM-AT has superior training efficiency but suffers from "**catastrophic overfitting**" problem [WRK20], i.e., the model lost the robust accuracy suddenly under strong adversarial attack such PGD attack and is overfitting to the weaker adversarial attack such FGSM attack, several works try to understand and mitigate the 'catastrophic overfitting' problem in order to achieve comparable adversarial robustness with PGD-AT in the training efficiency of FGSM-AT. Wong et al. [WRK20] propose to combine random initialization for the on-the-fly FGSM adversarial examples which enables FGSM-AT to achieve comparable performance as PGD-AT. Andriushchenko & Flammarion [AF20], and huang [HMPP20] further show that FGSM-AT with random initialization still suffers from the "catastrophic overfitting" problem and propose a regularization to mitigate this problem.

1.1.4 Evaluation Metrics

This thesis mainly focuses on two aspects of DNNs: Adversarial robustness and Task performance. Since most of the experiments in this study are conducted on the classification task, we use robust accuracy and clean accuracy to measure adversarial robustness and task performance respectively.
Formally, robust accuracy is expressed as follows:

$$\mathbb{E}_{(x,y)\sim D}\ \mathbf{1}\{arg\max_{i\in C} f_\theta^i(x+\delta)=y,\ \ \|\delta\|_p\leq\epsilon\} \tag{1.4}$$

In practice, δ is generated by adversarial attacks.
Correspondingly, clean accuracy is expressed as follows:

$$\mathbb{E}_{(x,y)\sim D}\mathbf{1}\{arg\max_{i\in C} f_\theta^i(x)=y\} \tag{1.5}$$

1.2 Research Questions

Adversarial examples play multiple roles in DNNs as shown in Figure 1.2. We define a number of research questions for each role (Figure 1.3) and express them as follows:

On the role of attacking DL models. Although adversarial examples in the white-box setting have been shown to fail DL models easily, the requirement of accessing target model knowledge hinders their application in practice. Besides, white-box attacks usually suffer from "gradient masking" effects, resulting in a false sense of robustness in evaluating adversarial robustness. Transferability in adversarial examples refers to the property of an adversarial attack, where the adversarial samples generated for one model can be effective in causing misclassification in other models, even if they have different architectures or are trained on different datasets. Understanding and improving the transferability of adversarial examples have great value for practical adversarial attacks and comprehensive evaluation of adversarial robustness. Therefore, we raise the first research question:

(RQ1) How to improve the transferability of adversarial examples?

On the role of building adversarial robust models. Currently, the most effective method to build adversarial robust models is to train models on on-the-fly adversarial examples (referred to as adversarial training technique). Although the adversarial training technique is effective in improving adversarial robustness, its performance is still far from satisfactory. It faces three problems: ❶ robust overfitting, ❷ a trade-off between clean accuracy and robust accuracy, ❸ expensive training cost. Studies delving into these three aspects are important to building models with more robust and efficient. Thus, the second research question raises resulting from the limitations described above:

(RQ2) Can we effectively address the three challenges: (1) Robust overfitting, (2) Trade-off between clean accuracy and robust accuracy, (3) High training cost?

On the role of boosting the generalization of DL models. Most of the existing studies mainly utilize adversarial examples for attacking DL models or building robust models in image classification problems. The roles of adversarial examples in representation learning are less explored. This exploration is of importance for the community to better understand and utilize adversarial examples. As a pilot study of the role of adversarial examples on representation learning, we raise the third research question:

(RQ3) Can adversarial examples enhance the representation learning of graph neural networks?

1.3 Methodology

(I) **To answer (RQ1)**

We study the transferability property of adversarial examples to improve their black-box attack ability. We propose to aggregate multiple attack directions sampled from the neighborhood of the input, leading to stable attack direction and avoiding overfitting to the specific white-box model. We evaluate the effectiveness of our proposed method by measuring the attack success rate of the generated adversarial examples on multiple unknown models including normal-trained and adversarial-trained models.

(II) **To answer (RQ2)**

We analyze adversarial training empirically and theoretically. Leveraging this analysis, we propose novel methods including (1) calibrated adversarial training for mitigating the trade-off between robust accuracy and clean accuracy, and (2) weighted optimization trajectories for eliminating the robust overfitting issue. Besides, we propose a novel regularization for reducing the training time of adversarial training and overcoming the catastrophic overfitting issue. To demonstrate the effectiveness of our proposed techniques, we conduct extensive experiments on multiple datasets including MNIST, SVHN, CIFAR-10, CIFAR-100, Tiny ImageNet with PreActResNet-18, and WideResNet-34 architectures. We present both clean and robust accuracy results under various adversarial attacks. Additionally, we perform ablation studies to demonstrate the impact of each component in our proposed method. To provide further insight, we also conduct visual analyses to illustrate the workings of our method.

(III) **To answer (RQ3)**

To preliminarily answer this question, we investigate adversarial training in the graph domain and prove its effectiveness in enhancing representation learning. We propose adversarial training for Graph Neural Networks (GNNs) and evaluate its performance across multiple tasks such as node classification, link prediction, and graph anomaly detection. Additionally, we examine the factors that influence the learned node representations.

1.4 Thesis Contribution and Outline

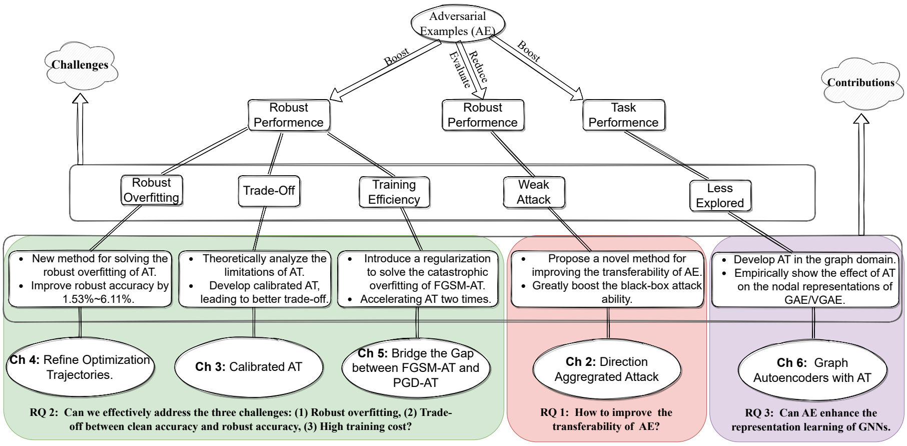

Adversarial Examples (AE)
Boost
Evaluate
Reduce
Boost
Challenges
Contributions
Robust Performence
Robust Performence
Task Performence
Robust Overfitting
Trade-Off
Training Efficiency
Weak Attack
Less Explored
New method for solving the robust overfitting of AT.
Improve robust accuracy by 1.53%~6.11%.
Theoretically analyze the limitations of AT.
Develop calibrated AT, leading to better trade-off.
Introduce a regularization to solve the catastrophic overfitting of FGSM-AT.
Accelerating AT two times.
Propose a novel method for improving the transferability of AE.
Greatly boost the black-box attack ability.
Develop AT in the graph domain.
Empirically show the effect of AT on the nodal representations of GAE/VGAE.
Ch 4: Refine Optimization Trajectories.
Ch 3: Calibrated AT
Ch 5: Bridge the Gap between FGSM-AT and PGD-AT
Ch 2: Direction Aggregrated Attack
Ch 6: Graph Autoencoders with AT
RQ 2: Can we effectively address the three challenges: (1) Robust overfitting, (2) Trade-off between clean accuracy and robust accuracy, (3) High training cost?
RQ 1: How to improve the transferability of AE?
RQ 3: Can AE enhance the representation learning of GNNs.

Figure 1.3: Contributions and Structures of this thesis.

We carry out a series of research studies during the course of this Ph.D. to delve into the above-mentioned research questions. These research questions are correlated with the core concept of adversarial examples (Figure 1.2 and Figure 1.3).

Firstly, we study the role of adversarial examples in attacking and evaluating a model. Concretely, we propose the **Direction-Aggregation** attack, which aggregates attack directions to reduce oscillation and prevent overfitting to the white-box model's decision boundary. Our experiments on the ImageNet dataset show that the direction-aggregation attack significantly improves the transferability of adversarial examples, making it a practical black-box attack. This study is an extension of the paper "*Direction-aggregated Attack for Transferable Adversarial Examples*" and corresponds to answer **RQ 1** (shown in Chapter 2).

Correspondingly, we study the role of adversarial examples in building robust models.

(1) We analyze the limitation of adversarial training and propose a new definition of robust error: **Calibrated Robust Error**. Besides, we derive an upper bound for the calibrated robust error. Furthermore, we propose calibrated adversarial training based on the upper bound of calibrated robust error, which can reduce the adverse effect of adversarial examples. Extensive experiments demonstrate that our method achieves the best performance on both clean and robust accuracy among baselines and provides a good trade-off between clean and robust accuracy. Furthermore, it enables training with larger perturbations, which yields higher adversarial robustness. This study is an extension of the paper "*Calibrated Adversarial Training*" and corresponds to answer **RQ 2** (reported in Chapter 3).

(2) Besides, to overcome the robust overfitting issue and improve robust generalization, we propose a new method named **Weighted Optimization Trajectories (WOT)** that leverages the optimization trajectories of adversarial training in time. We have conducted extensive experiments to demonstrate the effectiveness of WOT under various state-of-the-art adversarial attacks. Our results show that WOT integrates seamlessly with the existing adversarial training methods and consistently overcomes the robust overfitting issue, resulting in better adversarial robustness. This study is an extension of the paper "*In-Time Refining Optimization Trajectories Toward Improved Robust Generalization*" and corresponds to answer **RQ 2** (shown in Chapter 4).

(3) Moreover, we demonstrate that the large curvature along FGSM perturbed direction leads to a large difference in the performance of adversarial

robustness between FGSM-AT and PGD-AT, and therefore propose combining FGSM-AT with a curvature regularization in order to bridge the performance gap between FGSM-AT and PGD-AT. The experiments show that the proposed method achieves comparable adversarial robustness with PGD-AT but at the same training efficiency as FGSM-AT, which greatly accelerates the training of PGD-AT. This study is an extension of the paper "*Bridging the Performance Gap between FGSM and PGD Adversarial Training*" and corresponds to answer **RQ 2** (shown in Chapter 5).

Finally, we study the effect of adversarial training on the learned representations. We explore a case in the graph domain. Specifically, we first formulate L2 and L_∞ versions of adversarial training in two powerful node embedding methods: graph autoencoder (GAE) and variational graph autoencoder (VGAE). We experimentally show that both L2 and L_∞ adversarial training can boost the generalization with a large margin for the node embeddings learned by GAE and VGAE. The performance is highly influenced by the magnitude of adversarial perturbations and the strength of the regularization based on adversarial examples. This study is an extension of the paper "*On Generalization of Graph Autoencoders with Adversarial Training*" and corresponds to answer **RQ 3** (reported in Chapter 6).

Chapter 2

Direction-Aggregation for Transferable Adversarial Examples

Deep neural networks are vulnerable to adversarial examples that are crafted by imposing imperceptible changes to the inputs. However, these adversarial examples are most successful in white-box settings where the model and its parameters are available. Finding adversarial examples that are transferable to other models or developed in a black-box setting is significantly more difficult. This chapter proposes Direction-Aggregated adversarial attacks that deliver transferable adversarial examples. Specifically, our method utilizes the aggregated direction during the attack process for avoiding the generated adversarial examples overfitting to the white-box model. Extensive experiments on ImageNet show that our proposed method improves the transferability of adversarial examples significantly and outperforms state-of-the-art attacks, especially against adversarially trained models. The best-averaged attack success rate of our proposed method reaches 94.6% against three adversarially trained models and 94.8% against five defense methods. It also reveals that current defense approaches do not prevent transferable adversarial attacks.

2.1 Introduction

Deep Neural Networks (DNNs) have achieved great success in many tasks, e.g. image classification [KH12, HZRS16], object detection [GDDM14], segmentation [LSD15], etc. However, these high-performing models have been shown to be vulnerable to adversarial examples [SZS⁺13, lGS15]. In other words, carefully crafted changes to the inputs can change the model's prediction drastically. This fragility has raised concerns about security-sensitive tasks such as autonomous cars, face recognition, and malware detection. Well-designed adversarial examples are not only useful to evaluate the robustness of models against adversarial attacks but also beneficial to improve the model's robustness [lGS15].

Plenty of ways have been proposed to craft adversarial examples, which can be divided into white-box and black-box attacks. White-box attacks utilize complete knowledge including model architecture, model parameters, training strategy, and training method, e.g. fast gradient sign method (FGSM) [lGS15], Iterative Fast Gradient Sign Method (I-FGSM) [KGB17], Project gradient descent (PGD) [MMS⁺18], Deepfool [MDFF16], Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [DLP⁺18] and Carlini & Wagner's attack [CW17b]. On the contrary, black-box attacks fool the model's prediction without any knowledge about the model. It has been shown that adversarial examples generated by white-box attacks have the ability to fool other black-box models, which is known as the transferability property [SZS⁺13]. The transferability of adversarial examples enables practical black-box attacks and imposes a huge threat on real-world applications. However, the transferability of adversarial examples usually is very low because these adversarial examples easily overfit the white-box model, i.e. the model for generating these adversarial examples. Therefore, avoiding the *overfitting* problem is the key to generating transferable adversarial examples.

Deep neural networks applied to high dimensional classification tasks are typically very complex models, in other words, the decision boundary is highly non-linear and tends to have high curvature, e.g., the decision boundary of *model 1* in Fig. 2.1. We believe that it is the high curvature of a decision boundary that makes adversarial examples decrease their ability to attack other models, especially adversarial robust models [1] that have smoothed decision boundary [CRK19, LCWC19]. As shown in Fig. 2.1, the adversarial attack direction generated by model 1 at sample x (the black arrow line in Fig. 2.1) tends to overfit to model 1 because this attack direction is the best direction [2] for

[1] In this chapter, it denotes a model trained with an adversarial training technique.

[2] It denotes the direction that is perpendicular to the decision boundary.

attacking *model 1*, but not a good direction for attacking *model 2*. To mitigate the issue of adversarial examples easily overfitting to the white-box model, we propose to aggregate the attack directions from the neighborhood of the input x, e.g., by adding Gaussian noise or Uniform noise to the input. The green solid arrow line in Fig. 2.1 shows the aggregated direction. It is easy to see that the green solid arrow line is a good attack direction for both *model 1* and *model 2*. Therefore, adversarial examples generated by the aggregated direction can achieve good transferability. Based on this, we propose the Direction-Aggregated attack (DA-Attack) for improving the transferability of adversarial examples. Results of the extensive experiments presented in later sections show that our method achieves state-of-the-art results.

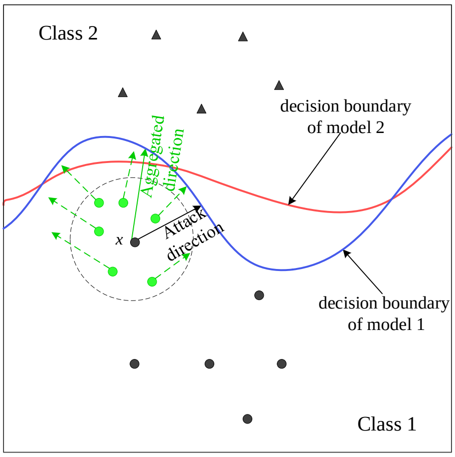


Figure 2.1: A simple schematic diagram for explaining why aggregated direction can mitigate the overfitting problem of adversarial examples. Black circle and triangle markers denote samples of class 1 and class 2 respectively. Red and blue lines represent the decision boundary of model 1 and model 2. The circle with a dotted line denotes a set of examples from the neighborhood of x. Black arrow line denotes the attack direction $(sgn(\nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x}), y)))$ of model 1 at the sample x. Green arrow dotted lines are the attack direction at the perturbed sample with Gaussian noise. Green arrow solid lines denote the aggregated direction by the vector addition of the green arrow dotted lines.

In detail, our contributions are summarized as follows:

- We propose to aggregate attack directions in order to stabilize the oscillation of attack directions and guide it to the generalized decision boundary and avoid overfitting to the white-box model's decision boundary. Based on the aggregated direction, we propose our DA-Attack.

- We demonstrate experimentally that DA-Attack outperforms state-of-the-art attacks through extensive experiments on ImageNet. The best-averaged attack success rate of our method achieves 94.6% against three ensemble adversarially trained models and 94.8% against five defense methods, which also reveals that current defense models are not safe for transferable adversarial attacks. We expect that the proposed DA-Attack will serve as a benchmark for evaluating the effectiveness of adversarial defense methods in the future.

- We experimentally show that sampling times N, standard deviation σ, iterations T, and perturbation size ϵ induced in our method plays an important role in achieving the transferability of adversarial examples. Usually, a bigger value in N, σ, T, and ϵ can lead to a higher transferability of the adversarial examples. However, a too-large value in T and σ would lead to a negative effect.

The rest of this paper is organized as follows. In Section 2.2 we present related work. In Section 2.3 we describe our proposed DA-Attack in detail. In Section 2.4 we discuss the results of the extensive experiments with DA-Attack. In Section 2.5 we discuss the connection of the DA-Attack to a smoothed classifier. We draw conclusions in Section 2.6.

2.2 Related Work

Adversarial examples Szegedy et al. [SZS+13] first found the existence of adversarial examples: given an input (x, y) and a classifier f_θ, it is possible to find a similar input x^* such that $f_\theta(x^*) \neq y$. A formal mathematical definition is as follows:

$$\min_{x^*} \|x^* - x\|_p, \;\; s.t.\;\; f_\theta(x^*) \neq y,\; f_\theta(x) = y \tag{2.1}$$

where $\|\cdot\|_p$ denotes the L_p distance.

Following [SZS+13], many related kinds of research have emerged. On the one hand, some of them propose to generate adversarial examples that can be applied in the physical world [EKM+18, KGB17]. On the other hand, some of them focus on reducing the minimal size of adversarial perturbations and improving the attack success rates [lGS15, DLP+18, CW17b, MDFF16]. Among these researches, the attack success rates under the black-box setting are still low, especially against adversarially trained models, i.e. the model is trained by adversarial training technique which can effectively defend against adversarial examples [MMS+18]. Recently, several papers improve the attack success rates based on transferable adversarial attacks. Inkawhich et al. [IWLC19] generate more transferable adversarial examples by enlarging the distance between adversarial examples and clean samples in feature space. Their intuition is from the fact that deep feature representations of models are transferable. Similar in utilizing feature representations, Zhou et al. [ZHC+18] improve the transferability by reducing the variations of adversarial perturbations via constructing a new regularization based on feature representations. Liu et al. [LCLS17] demonstrate that transferability can be improved by attacking an ensemble of substitute models. This method suffers from expensive computational costs since multiple models are needed to be trained first. Li et al. [LBZ+20] further reduce the computation cost of the method by attacking "Ghost Networks" where the "Ghost Networks" are generated from a basic trained model. Xie et al. [XZZ+19] believe that overfitting to the white-box model decreases the transferability of adversarial examples, therefore they induce the data augmentation technique to mitigate the overfitting issue. Specifically, they apply random transformations to the inputs and calculate gradients based on the transformed inputs. Dong et al. [DPSZ19] find that different models make predictions based on different discriminative regions of the input, which decreases the transferability of adversarial examples. Based on this intuition, they propose a translation-invariant attack by averaging the gradients from an ensemble of images composed of the image and its translated versions. Similarly, Lin et al. [LSH+20] enhance the transferability of adversarial examples by averaging gradients from an ensemble of images composed of the image and its scaled versions. Besides, Lin et al. [LSH+20] also demonstrate that Nesterov accelerated gradient can further improve the transferability of adversarial examples. Wu and Zhu [WZ20] improve the transferability of adversarial examples by smoothing the loss surface. Our method is degraded to this method when the attack direction of each step is the gradient of loss w.r.t the inputs. Naseer et al. [MSMH+19] propose "domain-agnostic" adversarial perturbations which can be used to fool models learned from different domains.

Defense against adversarial examples Correspondingly, many methods have been proposed to defend against these adversarial examples. Usually, the ability of a model for defending adversarial examples is referred to adversarial robustness. It measures a model's resilience against adversarial examples. Goodfellow, Shlen, and Szegedy [lGS15], Madry et al. [MMS+18] effectively improve a model's adversarial robustness by adversarial training technique. That is, it trains model based on on-the-fly generated adversarial examples x^* bounded by uniformly ϵ-ball of the input x (i.e., $\|x^* - x\| \leq \epsilon$). Tramer et al. [TKP+18] further improve adversarial robustness by ensemble adversarial training where the model is trained on adversarial examples generated from multiple pre-trained models. Cohen, Rosenfeld, and Kolter [CRK19] build a guaranteed adversarial robust model by transforming a base classifier f into a smoothed classifier's g. Specifically, the prediction of $g(X)$ is defined to be the class which f is most likely to classify the random variable $\mathcal{N}(x,\sigma^2 I)$ as. On the other hand, several papers try to defend against adversarial examples by purifying or reducing adversarial perturbations. Xie et al. [XWZ+18] and Guo et al. [GRCvdM18] impose transformations, e.g., image cropping, rescaling, quilting, padding, and so on, on input images at inference time to reduce the adversarial perturbations, and therefore increase the accuracy of the model's performance on adversarial examples. Liao et al. [LLD+18] propose a U-net based denoiser to purify the adversarial perturbations.

2.3 Methodology

In this section, we first introduce notation and then provide details of our method.

2.3.1 Notation

We specify the notations that are used in this chapter by the following list:

- $\boldsymbol{x}$ and y denote a clean image and the corresponding true label respectively.
- $\boldsymbol{x}^*$ denotes the adversarial example.
- $f_\theta(\boldsymbol{x})$ denotes a deep neural network.
- $L(f_\theta(\boldsymbol{x}), y)$ represents the Cross-Entropy loss.
- $sgn(\cdot)$ denotes the sign function.
- $\nabla_{\boldsymbol{x}} L(\cdot)$ denotes the gradient of $L(\cdot)$ with respect to $\boldsymbol{x}$.
- $Clip_{\boldsymbol{x}}^{\epsilon}(\cdot)$ function limits the generated adversarial example $\boldsymbol{x}^*$ to the ϵ max-norm ball of $\boldsymbol{x}$.
- ϵ is the allowed maximum perturbation size of the adversarial perturbation.

- α is the step size for PGD/FGSM-based adversarial attacks.
- $\mathcal{N}(0, \sigma^2 I)$ denotes Gaussian distribution with mean 0 and standard deviation σ.
- $U(a, b)$ is Uniform distribution.
- ε denotes a small random noise and can be generated from Gaussian distribution or Uniform distribution. In this chapter, we adopt Gaussian noise by default.
- $|\cdot|$ denotes the number of elements of a set.
- D^* denotes a set of adversarial examples.

2.3.2 Gradient-based Adversarial Attack Methods

Several adversarial attacks will be integrated into our proposed method. We give a brief introduction to them in this section.

Fast Gradient Sign Method (FGSM) [lGS15] generates adversarial examples by adding a fixed magnitude along the sign of gradients of the loss function, which is formalized as follows:

$$\boldsymbol{x}^* = \boldsymbol{x} + \epsilon \cdot sgn(\nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x}), y)). \tag{2.2}$$

Iterative Fast Gradient Sign Method (I-FGSM) [KGB16] is a multi-step variant of FGSM and restricts the perturbed size to the ϵ max-norm ball. With the initialization $\boldsymbol{x}_0^* = x$, the perturbed data in $t-th$ step $\boldsymbol{x}_t^*$ can be expressed as follows:

$$\boldsymbol{x}_t^* = Clip_{\boldsymbol{x}}^{\epsilon}\{\boldsymbol{x}_{t-1}^* + \alpha \cdot sgn(\nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x}_{t-1}^*), y))\}. \tag{2.3}$$

Momentum iterative fast gradient sign method (MI-FGSM) [DLP+18] integrates momentum into the I-FGSM method for stabilizing optimization, which can be expressed as follows:

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{\nabla \boldsymbol{x} L(f_\theta(\boldsymbol{x}_t^*), y)}{\|\nabla \boldsymbol{x} L(f_\theta(\boldsymbol{x}_t^*), y)\|_1} \tag{2.4}$$

$$\boldsymbol{x}_{t+1}^* = Clip_{\boldsymbol{x}}^{\epsilon}\{\boldsymbol{x}_t^* + \alpha \cdot sgn(\boldsymbol{g}_t)\} \tag{2.5}$$

where $\boldsymbol{g}_t$ is the accumulated gradient at iteration t and μ is the decay factor of the momentum term.

Diverse Inputs Method(DIM) [XZZ+19] calculates gradient based on random transformed inputs. The transformation contains random resizing and padding with a given probability. Formally, it can be expressed as follows:

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} L(f_\theta(T(\boldsymbol{x}_t^*; p)), y)}{\|\nabla_{\boldsymbol{x}} L(f_\theta(T(\boldsymbol{x}_t^*; p)), y)\|_1} \tag{2.6}$$

$$\boldsymbol{x}_{t+1}^* = Clip_{\boldsymbol{x}}^{\epsilon}\{\boldsymbol{x}_t^* + \alpha \cdot sgn(\boldsymbol{g}_{t+1})\} \tag{2.7}$$

where $T(\cdot; p)$ is the stochastic transformation function and p is the transformation probability.

Translation-invariant Method(TIM) [DPSZ19] generates an adversarial example by an ensemble of translated inputs and it was demonstrated to be equivalent to convolving the gradient at the untranslated image. Specifically, it can be expressed as follows:

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{\mathcal{W} * \nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x}_t^*), y)}{\|\mathcal{W} * \nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x}_t^*), y)\|_1} \tag{2.8}$$

$$\boldsymbol{x}_{t+1}^* = Clip_{\boldsymbol{x}}^{\epsilon}\{\boldsymbol{x}_t^* + \alpha \cdot sgn(\boldsymbol{g}_t)\} \tag{2.9}$$

where $*$ is the convolutional operation and $\mathcal{W}$ is the kernel matrix of size $(2k+1) \times (2k+1)$. Following [DPSZ19], a Gaussian kernel is chosen for our experiments. It is defined as: $\tilde{\mathcal{W}}_{i,j} = \frac{1}{2\pi\sigma^2} \exp - \frac{i^2+j^2}{2\sigma^2}$ where the standard deviation $\sigma = k/\sqrt{3}$ and $\mathcal{W}_{i,j} = \tilde{\mathcal{W}}_{i,j} / \sum_{i,j} \tilde{\mathcal{W}}_{i,j}$.

2.3.3 Direction-Aggregated Attack (DA-Attack)

In Fig. 2.1 we illustrated that adversarial examples could overfit to the white-box model due to the very complex decision boundary decreasing their transferability. We mitigate this overfitting problem by aggregating the attack directions of a set of examples from the neighborhood of the input. We integrate the aggregated direction to basic adversarial attacks, i.e. Fast Gradient Sign Method (FGSM) [SZS+13], Iterative Fast Gradient Sign Method (I-FGSM) [KGB16], and Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [DLP+18], for improving their transferability. Besides, to further enhance the transferability, we combine our method with other transferable adversarial attacks, i.e. Diverse Input Method (DIM) [XZZ+19], Translation-Invariant Method (TIM) [DPSZ19], TI-DIM [DPSZ19]. Concretely, the update procedures for each attack are formalized as follows.

DA-FGSM. To mitigate the effect of overfitting to the specific model and improve the transferability of adversarial examples for FGSM attack, we propose the Direction-Aggregated FGSM (DA-FGSM). The attack direction is replaced with the aggregated direction which is achieved by aggregating the attack directions of a set of examples from the neighborhood of the input x. In practice, we generate the set of examples by adding small perturbations to the input, i.e. adding Gaussian noise or Uniform noise to the input. In this chapter, we adopt Gaussian noise as the default choice. We further provide evidence that Uniform noise can reach the same performance as Gaussian noise. Formally, it can be represented as follows:

$$\boldsymbol{x}^* = \boldsymbol{x} + \epsilon \cdot sgn(\sum_{i=0}^{N} (sgn(\nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x} + \varepsilon_i), y)))), \tag{2.10}$$

where N denotes the sampling times from certain noise distribution. The $sgn(\nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x} + \varepsilon_i), y))$ denotes one specific attack direction. We aggregate the N attack directions by the sum operation.

DA-I-FGSM. To improve the transferability for I-FGSM. We propose the Direction-Aggregated I-FGSM (DA-I-FGSM). The attack direction at each iteration is replaced with the aggregated direction. The update procedure can be formalized as follows:

$$\boldsymbol{x}_t^* = Clip_{\boldsymbol{x}}^{\epsilon}\{\boldsymbol{x}_{t-1}^* + \alpha \cdot sgn(\sum_{i=0}^{N} (sgn(\nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x}_{t-1}^* + \varepsilon_i), y))))\}. \tag{2.11}$$

DA-MI-FGSM. We integrate the momentum term into DA-I-FGSM for improving the attack ability, which is called Momentum Direction-Aggregated I-FGSM (DA-MI-FGSM). The update procedure of DA-MI-FGSM can be expressed as follows:

$$\boldsymbol{g}_a = \sum_{i=0}^{N} (sgn(\nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x}_{t-1}^* + \varepsilon_i), y))) \tag{2.12}$$

$$\boldsymbol{g}_t = \mu \cdot \boldsymbol{g}_{t-1} + \frac{\boldsymbol{g}_a}{\|\boldsymbol{g}_a\|_1} \tag{2.13}$$

$$\boldsymbol{x}_t^* = Clip_{\boldsymbol{x}}^{\epsilon}\{\boldsymbol{x}_{t-1}^* + \alpha \cdot sgn(\boldsymbol{g}_t)\}, \tag{2.14}$$

where $\boldsymbol{g}_t$ is the accumulated gradient at iteration t and μ is the decay factor of the momentum term, and $\boldsymbol{g}_a$ is the aggregated direction.

DA-DIM. We combine our proposed DA-MI-FGSM with DIM to further improve the transferability of adversarial examples and denote it as Direction-Aggregated DIM (DA-DIM). The update procedure is similar to DA-MI-FGSM, with the replacement of Eq. (2.12) by the following equation:

$$\boldsymbol{g}_a = \sum_{i=0}^{N} (sgn(\nabla_{\boldsymbol{x}} L(f_\theta(T(\boldsymbol{x}^*_{t-1} + \varepsilon_i; p)), y))), \tag{2.15}$$

DA-TIM. Similar to DA-DIM, we combine DA-MI-FGSM with TIM and denote it as Direction-Aggregated TIM (DA-TIM). Likewise, the update procedure is similar to DA-MI-FGSM, with the replacement of Eq. (2.13) by the following equation:

$$\boldsymbol{g}_t = \mu \cdot \boldsymbol{g}_{t-1} + \frac{\mathcal{W} * \boldsymbol{g}_a}{\|\mathcal{W} * \boldsymbol{g}_a\|_1}, \tag{2.16}$$

DA-TI-DIM. Following [LSH+20], we combine DA-MI-FGSM with TIM and DIM together and denote it as Direction-Aggregated TI-DIM (DA-TI-DIM). The update procedure can be presented as follows:

$$\boldsymbol{g}_a = \sum_{i=0}^{N} (sgn(\nabla_{\boldsymbol{x}} L(f_\theta(T(\boldsymbol{x}^*_{t-1} + \varepsilon_i; p)), y))) \tag{2.17}$$

$$\boldsymbol{g}_t = \mu \cdot \boldsymbol{g}_{t-1} + \frac{\mathcal{W} * \boldsymbol{g}_a}{\|\mathcal{W} * \boldsymbol{g}_a\|_1} \tag{2.18}$$

$$\boldsymbol{x}^*_t = Clip^{\epsilon}_{\boldsymbol{x}}\{\boldsymbol{x}^*_{t-1} + \alpha \cdot sgn(\boldsymbol{g}_t)\}. \tag{2.19}$$

The pseudocode of DA-MI-FGSM is summarized in Algorithm 1 and the code is provided[3].

2.4 Experiments

We evaluate the effectiveness of DA-Attack empirically. We first introduce the dataset and experimental settings. Then we show the performance of our method against normal and defense models. Finally, we analyze the influence of the parameters N, σ, ϵ, T and α on achieving the transferability of adversarial examples.

2.4.1 Experimental Settings

Datasets. Following the strategy used in [LSH+20], a set of 1000 images (denoted as D) that are correctly classified by all testing models are randomly

[3] `https://github.com/Juintin/DA-Attack.git`

Algorithm 1 DA-MI-FGSM

```
Require: A input image $x$ with true label $y$; a classifier $f$ with loss function $L$;
    perturbation size $\epsilon$; maximum iterations $T$; Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$;
    The decay factor $\mu$; the aggregated direction $\boldsymbol{g_a}$.
Ensure: An adversarial example $\boldsymbol{x}^*$
$\alpha = \epsilon/T$
$\boldsymbol{x}_0^* = \boldsymbol{x}$; $\boldsymbol{g}_0 = 0$
for $t = 1$ to $T$ do
    $\boldsymbol{g_a} = 0$
    for $i = 0$ to $N$ do
        Get $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I)$
        Aggregate attack directions as $\boldsymbol{g}_a = \boldsymbol{g}_a + sgn(\nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x}_{t-1}^* + \varepsilon_i), y))$
    end for
    Update $\boldsymbol{g}_t = \mu \cdot \boldsymbol{g}_{t-1} + \frac{\boldsymbol{g}_a}{\|\boldsymbol{g}_a\|_1}$
    Update $\boldsymbol{x}_t^* = Clip_x^\epsilon\{\boldsymbol{x}_{t-1}^* + \alpha \cdot sgn(\boldsymbol{g}_t)\}$
end for
$\boldsymbol{x}^* = \boldsymbol{x}_t^*$
return $\boldsymbol{x}^*$
```

selected from ILSVRC 2012 validation set. For a fair comparison with state-of-the-art methods, we use the same 1000 images[4] in [LSH+20].

Models. Four normally trained models and three ensemble adversarially trained models are used for evaluating adversarial examples, which are Inception-V3 (Inc-V3) [SVI+16], Inception-v4 (Inc-V4) [SIVA17], Inception-Resnet-v2 (IncRes-V2) [SIVA17], Resnet-V2 (Res-101) [HZRS16], Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ [TKP+18] respectively. Besides, five advanced defense methods are considered for further evaluating the effectiveness of our method. Specifically, the selected advanced defense methods are High-level representation guided denoiser (HGD) [LLD+18], Random resizing and padding (R&P) [XWZ+18], NIPS-r3[5], feature distillation (FD) [LLL+19] and purifying perturbations by image compression (Comdefend) [JWCF19].

Baselines. Several most recently proposed methods aiming at generating transferable adversarial examples are taken as baselines:

- DIM [XZZ+19], which generates transferable examples by random resizing input images;

[4] `https://github.com/JHL-HUST/SI-NI-FGSM`

[5] `https://github.com/anlthms/nips-2017/tree/master/mmd`

- TIM [DPSZ19], which generates transferable examples by a set of translated images;
- SI-NI-FGSM [LSH$^+$20], which generates transferable examples by scaled images and Nesterov accelerated gradients; and
- The combinations of DIM, TIM, and SI-NI-FGSM, namely TI-DIM [DPSZ19], SI-NI-TIM [LSH$^+$20], SI-NI-DIM [LSH$^+$20] and SI-NI-TI-DIM [LSH$^+$20] attacks.

Considering that we completely follow the experimental settings in [LSH$^+$20], all the baseline results except for the attack success rates against FD and ComDefend in Table 2.6 are from [LSH$^+$20].

Hyper-Parameters. We follow the settings in [LSH$^+$20] for all hyper-parameters, the maximum perturbation ϵ is set to 16, and the number of iterations T is set to 12 as default values. Accordingly $\alpha = \epsilon / T$. The momentum parameter μ is set to 1.0. For DIM and TI-DIM methods, the transformation probability is set to 0.5. For the TIM method, the Gaussian kernel is adopted as our baseline experiments and kernel size is set to 7×7. For SI-NI-FGSM, SI-NI-TIM, SI-NI-DIM, and SI-NI-TI-DIM methods, the number of scales is set to 5. For our DA-Attack, sampling times N and standard deviation σ are set to 30 and 0.05 respectively.

Criteria. We use the attack success rates to reflect the ability of adversarial examples to attack a model. The attack success rates are defined as follows:

$$100 \times \frac{\sum_{i=1}^{M} [\arg\max_j f_j(x_i^*) \neq y_i]}{M}, \tag{2.20}$$

where $(x_i^*, y_i) \in D^*$ and M is the number of adversarial examples in D^*.

2.4.2 Single-Model Attacks

We first evaluate the effectiveness of DA-Attack based on the single model. DIM [XZZ$^+$19], TIM [DPSZ19] and SI-NI-FGSM [LSH$^+$20] and their combinations, i.e. SI-NI-TIM, TI-DIM, SI-NI-TI-DIM, are taken as baselines. Besides, several popular normal adversarial attacks, i.e. FGSM, I-FGSM, MI-FGSM, PGD, C&W, are utilized to show the effectiveness of our method.

Comparison with normal and transferable attacks. The attack success rates of DIM, TIM, SI-NI-FGSM, normal attacks and our proposed method are shown in Table 2.1. The adversarial examples are crafted based on the Inc-V3 model. From Table 2.1, it can be observed:

- Adversarial examples are much easier to attack normally trained models than adversarially trained models.

- Adversarial examples generated by transferable attacks have much higher attack success rates against black-box models than normal attacks.

- Our proposed M-ADI-FGSM attack outperforms the current state-of-the-art SI-NI-FGSM attack by 4.6% to 10.4%. Besides, DA-FGSM and DA-I-FGSM attacks without momentum acceleration still achieve remarkable results compared with normal attacks, which demonstrates the effectiveness of the aggregated direction.

Besides, it is worth noting that adversarial examples from the I-FGSM attack are less transferable than that from the FGSM attack (by comparing I-FGSM with FGSM in Table 2.1), which shows the evidence that adversarial examples overfitting to the white-box model decreases the transferability. And the transferability is improved by adding a momentum term during generating adversarial examples (by comparing MI-FGSM with I-FGSM in Table 2.1), which is in line with the claim in [DLP+18]. Interestingly, the combination of Direction Aggregation and momentum can greatly improve the transferability again (by comparing MI-FGSM with DA-MI-FGSM in Table 2.1). We conjecture that it is because the proposed Direction Aggregation technique is orthogonal to the momentum technique. Intuitively, the Direction Aggregation technique stabilizes the attack direction by reducing the oscillation of each update direction during the iterations while momentum stabilizes the attack direction by accumulating historical update directions.

Comparison with the extensions of DIM and TIM. To fully evaluate DA-TIM, DA-DIM, and DA-TI-DIM attacks, adversarial examples are crafted by these attacks based on Inc-V3, Inc-V4, IncRes-V2, and Res-101 models respectively. We test it against the four normally trained and three ensemble adversarially trained models. The evaluation results are shown in Table 2.2, Table 2.3 and Table 2.4. It can be observed from these results:

- The combination of our method with DIM and TIM methods significantly enhances the transferability of adversarial examples, demonstrating that our method is complementary to these methods.

- Our method outperforms the state-of-the-art attacks across all conducted experiments, i.e. SI-NI-TIM, SI-NI-DIM, and SI-NI-TI-DIM, except for adversarial examples crafted on IncRes-V2 model. Besides, the attack success rates of our method against the adversarially trained models outperform state-of-the-art attacks by large margins.

For the exception that our method does not outperform the state-of-the-art results for adversarial examples crafted on the IncRes-V2 model, it may be because the adversarial examples generated by our method underfit the IncRes-V2 model somehow since the attack success rates for the white-box model IncRes-V2 is only around 95% and 4%-5% lower than the SI-NI-TIM/DIM method. One possible solution for this "underfit" problem is to increase the Iterations T. The results in Fig. 2.7c also indicates that the attack success rates for normal models can be improved a lot by increasing the Iterations T. Besides, we notice that the improvement of combining DA technique and DIM/TIM implemented on different white-box models are different. We think it may be caused by the different degrees of non-linearity on the decision boundaries of different white-box models. Intuitively, the greater the non-linearity of the decision boundary, the larger the improvement in transferability that can be achieved with the DA technique.

Visibility. We visualize 5 randomly selected pairs of adversarial examples generated by TIM, DIM, SI-NI-FGSM, and DA-MI-FGSM attacks respectively, and their corresponding clean images in Fig. 2.2. We can see that the adversarial examples generated by our method are similar to those generated by other methods in visibility, and all these adversarial examples are hard to be distinguished from their corresponding clean images by humans.

Table 2.1: The attack success rates (%) against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models. The adversarial examples are generated based on the Inc-V3 model by normal adversarial attacks including FGSM, I-FGSM, PGD, C&W, and transferable adversarial attacks including DIM, TIM, SI-NI-FGSM, DA-FGSM, DA-I-FGSM, and DA-MI-FGSM attacks. $*$ denotes the white-box model being attacked.

	Attack	Inc-V3*	Inc-V4	IncRes-V2	Res-101	Inc-V3$_{ens3}$	Inc-V3$_{ens4}$	IncRes-V2$_{ens}$
Normal	FGSM	67.1	26.7	25	24.4	10.5	10	4.5
	I-FGSM	99.9	20.7	18.5	15.3	3.6	5.8	2.9
	PGD	99.5	17.3	15.1	13.1	6.1	5.6	3.1
	C&W	100	18.4	16.2	14.3	3.8	4.7	2.7
Transferable	MI-FGSM	100.0	40.0	38.2	32.3	12.5	12.8	6.8
	DIM	98.7	67.7	62.9	54	20.5	18.4	9.7
	TIM	100	47.8	42.8	39.5	24	21.4	12.9
	SI-NI-FGSM	100	76	73.3	67.6	31.6	30	17.4
	DA-FGSM(Ours)	87.6	47	43.6	42	18.3	17.4	9.5
	DA-I-FGSM(Ours)	99.8	44	39.2	34.3	23.7	22.4	12.4
	DA-MI-FGSM(Ours)	99.8	**80.6**	**78.5**	**72.2**	**40.6**	**40.4**	**26.5**

Table 2.2: Comparison of TIM, SI-NI-TIM and the DA-TIM extension. The attack success rates (%) are shown in the table. Adversarial examples are generated based on Inc-V3, Inc-V4, IncRes-V2, and Res-101 respectively. $*$ denotes the attack success rates under white-box attacks.

Model	Attack	Inc-V3	Inc-V4	IncRes-V2	Res-101	Inc-V3$_{ens3}$	Inc-V3$_{ens4}$	IncRes-V2$_{ens}$
Inc-V3	TIM	100*	47.8	42.8	39.5	24	21.4	12.9
	SI-NI-TIM	100*	77.2	75.8	66.5	51.8	45.9	33.5
	DA-TIM(Ours)	99.8*	**80.9**	**77.9**	**71.8**	**66.9**	**65.2**	**51.2**
Inc-V4	TIM	58.5	99.6*	47.5	43.2	25.7	23.3	17.3
	SI-NI-TIM	83.5	100*	76.6	68.9	57.8	54.3	42.9
	DA-TIM(Ours)	**84.2**	98.4*	**77.7**	**69.3**	**66.8**	**65.9**	**56.4**
IncRes-V2	TIM	62	56.2	97.5*	51.3	32.8	27.9	21.9
	SI-NI-TIM	**86.4**	**83.2**	99.5*	**77.2**	66.1	60.2	57.1
	DA-TIM(Ours)	80	78.5	94*	74	**69.5**	**66.4**	**66**
Res-101	TIM	59	53.6	51.8	99.3*	36.8	32.2	23.5
	SI-NI-TIM	78.3	74.1	73	99.8*	58.9	53.9	43.1
	DA-TIM(Ours)	**78.6**	**74.7**	**76**	99.2*	**72.1**	**69.7**	**62.7**

2.4.3 Ensemble-based Attacks

We also evaluate the performance of our method under ensemble-based attacks. Liu et al. [LCLS17] have shown that attacking multiple models simultaneously can generate more transferable adversarial examples. It is because if an adversarial example can attack multiple models successfully, it can more likely attack yet another model successfully.

We follow the ensemble-based attack strategy proposed in [DLP+18], which fuses the logit activations of multiple models to generate adversarial examples. In this experiment, we generate adversarial examples by attacking Inc-V3, Inc-V4, IncRes-V2, and Res-101 models simultaneously with equal ensemble weights. In Table 2.5, we show the attack success rates for DA-DIM, DA-TIM, DA-TI-DIM, and baselines. It shows that our method outperforms these baselines across all experiments. The highest attack success rate is achieved by our DA-TI-DIM attack and the average attack success rates against the three robust models reach 94.6%.

2.4.4 Attacking Other Defense Models

We also study the performance of our method on defense models. We test it against HGD [LLD+18], R&P [XWZ+18], NIPS-r3, FD [LLL+19] and ComDefend [JWCF19] defense methods. HGD, R&P, and NIPS-r3 were the top 3 defense methods in the NIPS 2017 defense competition. FD and ComDefend are recently published defense methods for purifying adversarial perturbations. TI-DIM [DPSZ19] and SI-NI-TI-DIM attacks [LSH+20] are presented as base-

Table 2.3: Comparison of DIM, SI-NI-DIM and the DA-DIM extension. The numbers in the table denote the attack success rates (%). Adversarial examples are generated based on Inc-V3, Inc-V4, IncRes-V2, and Res-101 respectively using DIM, SI-NI-DIM, and DA-DIM methods. $*$ denotes the attack success rates under white-box attacks.

Model	Attack	Inc-V3	Inc-V4	IncRes-V2	Res-101	Inc-V3$_{ens3}$	Inc-V3$_{ens4}$	IncRes-V2$_{ens}$
Inc-V3	DIM	98.7*	67.7	62.9	54	20.5	18.4	9.7
	SI-NI-DIM	99.6*	84.7	81.7	75.4	36.9	34.6	20.2
	DA-DIM(Ours)	99.5*	**89**	**87.3**	**81.2**	**57.1**	**56.6**	**38.8**
Inc-V4	DIM	70.7	98.0*	63.2	55.9	21.9	22.3	11.9
	SI-NI-DIM	89.7	99.3*	84.5	78.5	47.6	45	28.9
	DA-DIM(Ours)	**90.8**	98.1*	**87.1**	**80.9**	**62.1**	**62.9**	**49.7**
IncRes-V2	DIM	69.1	63.9	93.6*	47.4	29.4	24	17.3
	SI-NI-DIM	**89.7**	**86.4**	99.1*	**81.2**	55	48.2	38.1
	DA-DIM(Ours)	86.1	85.8	95*	80.2	**64.6**	**59.7**	**57.1**
Res-101	DIM	75.9	70	71	98.3*	36	32.4	19.3
	SI-NI-DIM	88.7	84.2	84.4	99.3*	53.4	48	33.2
	DA-DIM(Ours)	**90.9**	**87.7**	**89.4**	99.2*	**75.3**	**72.6**	**62.9**

lines. Adversarial examples are generated based on the ensemble of Inc-V3, Inc-V4, IncRes-V2, and Res-101 models. The attack success rates against FD and ComDefend defense are based on IncRes-V2$_{ens}$ model.

As shown in Table 2.6, our model achieves state-of-the-art results and reaches 94.8% for averaged attack success rates, which indicates current defense methods are not safe to transferable adversarial attacks.

2.4.5 Similarity of Adversarial Perturbations

To further understand the proposed Direction-Aggregated attack, we plot the cosine similarity of adversarial perturbations generated from multiple white-box models, i.e. Inc-V3, Inc-V4, IncRes-V2, and Res-101 models. The results are shown in Fig 2.3.

In Fig 2.3, the cosine similarity of adversarial perturbations generated by the proposed Direction-Aggregated attack is generally higher than other baseline attacks. It is in line with our expectation since the aggregated direction could reduce the oscillation of each update direction in generating adversarial perturbations. Besides, we notice that the cosine similarity of adversarial perturbations on DA-FGSM is not significantly higher than FGSM. We conjecture that it is due to the adversarial perturbations generated by FGSM "underfit" the white-box model, which limits the similarity of adversarial perturbations.

Table 2.4: Comparison of TI-DIM, SI-NI-TI-DIM and the DA-TI-DIM extension. The numbers in the table denote the attack success rates (%). Adversarial examples are generated based on Inc-V3, Inc-V4, IncRes-V2, and Res-101 respectively using TI-DIM, SI-NI-TI-DIM, and DA-TI-DIM methods. $*$ denotes the attack success rates under white-box attacks.

Model	Attack	Inc-V3	Inc-V4	IncRes-V2	Res-101	Inc-V3$_{ens3}$	Inc-V3$_{ens4}$	IncRes-V2$_{ens}$
Inc-V3	TI-DIM	98.5*	66.1	63	56.1	38.6	34.9	22.5
	SI-NI-TI-DIM	99.6*	85.5	80.9	75.7	61.5	56.9	40.7
	DA-TI-DIM(Ours)	99.6*	**88.3**	**85.1**	**80.3**	**77.4**	**76.8**	**62.9**
Inc-V4	TI-DIM	72.5	97.8*	63.4	54.5	38.1	35.2	25.3
	SI-NI-TI-DIM	88.1	99.3*	83.7	77	65	63.1	49.4
	DA-TI-DIM(Ours)	**88.8**	97.8*	**83.9**	**78.3**	**75.7**	**75.7**	**68.1**
IncRes-V2	TI-DIM	73.2	67.5	92.4*	61.3	46.4	40.2	35.8
	SI-NI-TI-DIM	**89.6**	**87**	99.1*	**83.9**	74	67.9	63.7
	NS-TI-DIM(Ours)	84.2	83.5	94.5*	78.3	**76.1**	**73.1**	**72.8**
Res-101	TI-DIM	74.9	69.8	70.5	98.7*	52.6	49.1	37.8
	SI-NI-TI-DIM	86.4	82.6	84.6	99*	72.6	66.8	56.4
	DA-TI-DIM(Ours)	**88.1**	**83.8**	**86.2**	99.3*	**82.6**	**82.2**	**76.2**

2.4.6 Parameter Analysis

In this section, we conduct a series of experiments to study the impact of different hyper-parameters on the transferability of adversarial examples.

Sampling Times N**.** We explore the influence of sampling times N upon the transferability of adversarial examples. Fig. 2.4 shows the attack success rates (%) against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models under black-box settings. The generation of adversarial examples is based on Inc-V3, Inc-V4, IncRes-V2, and Res-101 models respectively with standard deviation σ setting as 0.05.

From Fig. 2.4, we can see that the attack success rates are growing with the increase in sampling times. In detail, the curve is growing fast when sampling times N are less than 30 and the trend of growth tends to be flattening when sampling times N are greater than 30. Besides, the growing trends of Fig. 2.4a, Fig. 2.4b, Fig. 2.4c and Fig. 2.4d are similar, which indicates that the influence of sampling times N on the transferability is little sensitive to the white-box model.

σ **in Gaussian Distribution.** Standard deviation σ controls the shape of Gaussian distribution and plays an important role in Gaussian noise generation. We study the influence of σ upon the transferability of adversarial examples. Fig. 2.5 shows the attack success rates against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models under black-box attacks. Adversarial examples in this experiment are crafted based on Inc-V3, Inc-V4, IncRes-V2, and Res-101 models respectively with sampling times $N = 30$.

From Fig. 2.5, we can see that the attack success rates have a surge increasing

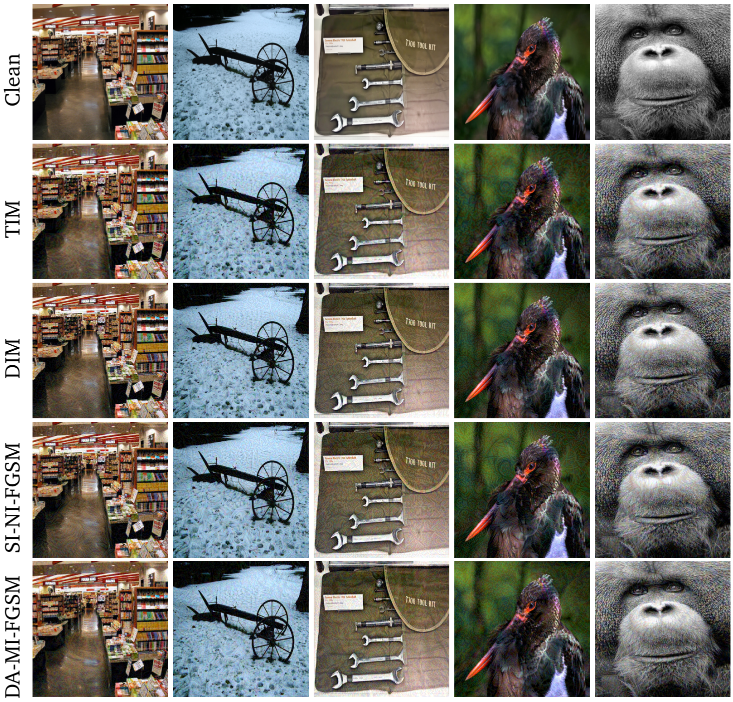


Figure 2.2: Visualization of randomly selected clean images and their corresponding adversarial examples. All examples are generated by TIM, DIM, SI-NI-FGSM, and DA-MI-FGSM attacks respectively.

at first, then the growing trends tend to be flattening. The surge increasing of the attack success rates indicates that the parameter σ plays an important role in our method. Besides, the similar trends among Fig. 2.5a, Fig. 2.5b, Fig. 2.5c and Fig. 2.5d indicate that the influence of σ on achieving transferability is insensitive to the white-box model.

It is deserved to note that a very large σ is not encouraged for our method for two reasons: 1) a larger σ indicates a larger perturbation size will be generated (Fig. 2.1), thus more sampling times are needed to cover the sampling region; 2) noise sampling from a very large σ might already be too large to flip the prediction and consequently disturb the attack direction.

Perturbation Size ϵ. We study the impact of perturbation size ϵ on the attack success rates. We set sampling times N and standard deviation σ to 30 and

Table 2.5: The attack success rates (%) against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models. Adversarial examples are generated based on the ensemble of Inc-V3, Inc-V4, IncRes-V2, and Res-101 models using DIM, SI-NI-DIM, TIM, SI-NI-TIM, TI-DIM, SI-NI-TI-DIM, DA-DIM, DA-TIM, and DA-TI-DIM attacks respectively. $Average$ column denotes the averaged attack success rates against the three robust models. $*$ denotes the white-box model being attacked.

Attack	Inc-V3*	Inc-V4*	IncRes-V2*	Res-101*	Inc-V3$_{ens3}$	Inc-V3$_{ens4}$	IncRes-V2$_{ens}$	Average
DIM	99.7	99.2	98.9	98.9	66.4	60.9	41.6	56.3
SI-NI-DIM	100	100	100	99.9	88.2	85.1	69.7	81
DA-DIM(Ours)	99.9	99.8	99.7	99.8	**91**	**90.1**	**85.5**	**88.9**
TIM	99.9	99.3	99.3	99.8	71.6	67	53.2	63.9
SI-NI-TIM	100	100	100	100	93.2	90.1	84.5	89.2
DA-TIM(Ours)	99.8	99.8	99.2	99.6	**93.4**	**92.1**	**89.3**	**91.6**
TI-DIM	99.6	98.8	98.8	98.9	85.2	80.2	73.3	79.5
SI-NI-TI-DIM	99.9	99.9	99.9	99.9	96	94.3	90.3	93.5
DA-TI-DIM(Ours)	99.8	99.8	99.6	99.6	**96.2**	**94.7**	**93**	**94.6**

Table 2.6: The attack success rates against the five advanced defense models.

Attack	HGD	R&P	NIPS-r3	FD	ComDefend	Average
TI-DIM	84.8	75.3	80.7	84.2	79.6	80.9
SI-NI-TI-DIM	96.1	91.3	94.4	93.7	91.9	93.5
DA-TI-DIM(Ours)	**96.1**	**93.6**	**94.8**	**94.4**	**94.3**	**94.8**

0.05 respectively. We fix step size $alpha$ to $\frac{16}{10}$ and iterations T to 16. The attack success rates (%) against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models are achieved under black-box settings. The ϵ varies from 10 to 16 and the results are shown in Fig. 2.6.

From Fig. 2.6, we observe that the attack success rates increase steadily as perturbation size ϵ increases on both adversarially trained models and normally trained models.

Iterations T. We study the impact of iterations T on the transferability of adversarial examples. Similarly, we set sampling times N and standard deviation σ to 30 and 0.05 respectively. We fix perturbation size ϵ to 16 and step size α to $\frac{16}{10}$. We generate adversarial examples based on normally trained models. Then these adversarial examples are tested on the other models under black-box settings. The total iterations T vary from 5 to 22 and the results are shown in Fig. 2.7.

From Fig. 2.7, we can see that the attack success rates are growing significantly when T is less than 10. However, the attack success rates start to flatten/slightly grow on the normally trained models and slightly decrease on the adversarially trained models after T is greater than 10. It is worth noting

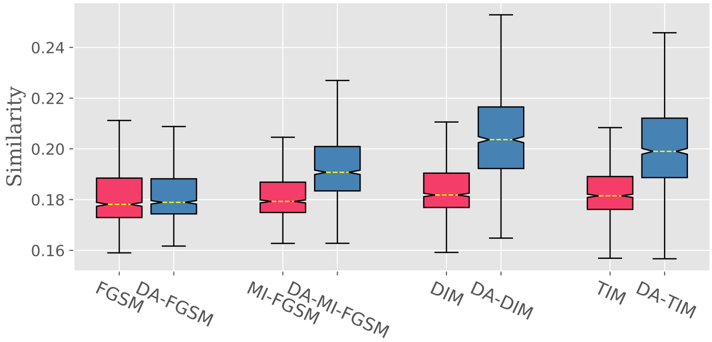
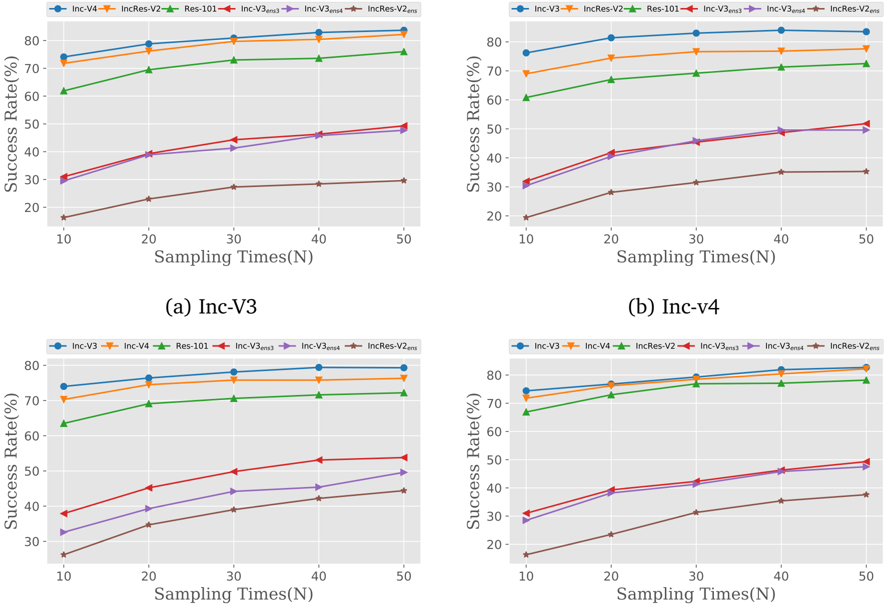


Figure 2.3: The cosine similarity of adversarial perturbations generated from Inc-V3, Inc-V4, IncRes-V2 and Res-101 models.

that the perturbation size reaches the maximum perturbation size because the α is set to $\frac{16}{10}$, which could be the reason why the trends start to be flattening after $T = 10$. Besides, we conjecture that the adversarial examples overfit the white-box model to some extent when T is greater than 10, which decreases its transferability. A similar phenomenon can be found in $I - FGSM$ attack in which the adoption of multiple iterations decreases its transferability. A possible reason for the steady/slight increase in the normally trained models when $T > 10$ is that the decision boundary of the white-box model is more similar to that of the normally trained models than that of the adversarially trained models.

Step size α. We study the impact of step size α on the transferability of adversarial examples. Similarly, we set sampling times N and standard deviation σ to 30 and 0.05 respectively. We fix perturbation size ϵ to 16 and iterations T to 16. We generate adversarial examples based on normally trained models. Then we test these adversarial examples on the other models under black-box settings. The step size α varies from $\frac{16}{8}$ to $\frac{16}{16}$ and the results are shown in Fig. 2.8.

From Fig. 2.8, it can be seen that the attack success rates are consistently increasing with the decrease of α on the adversarially trained models while keeping a flat/slightly decreasing trend on normally trained models. The reason for the different trends between normally trained models and adversarially trained models might be because the correctly classified samples by normally trained models are very difficult to conduct the transferable attack. To show the evidence for our conjecture, we provide a ratio metric to indicate the percentage of the samples correctly classified by normal models are also correctly classified



(a) Inc-V3

(b) Inc-v4

(c) IncRes-V2

(d) Res-101

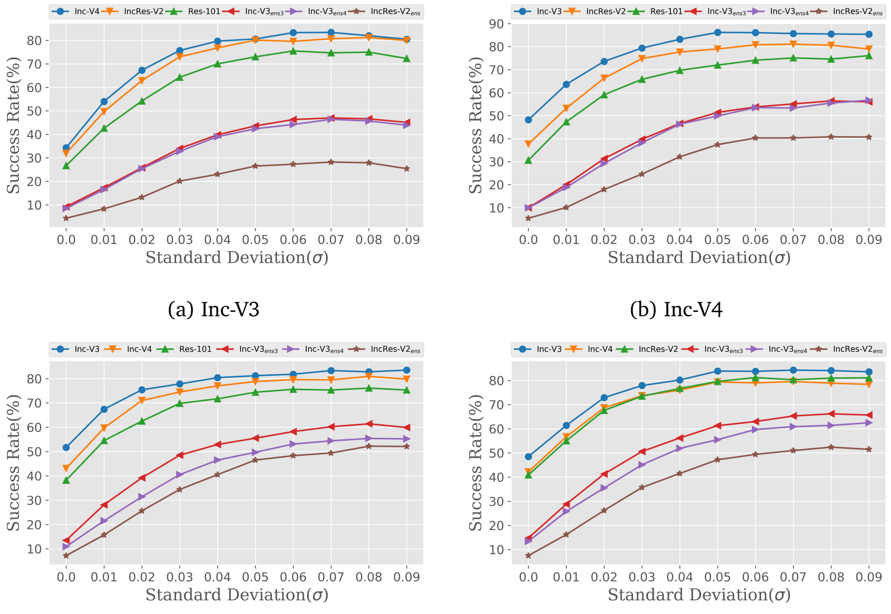
Figure 2.4: The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models when varying sampling times N ranging from 10 to 50. The adversarial examples are generated based on Inc-V3 (Fig. 2.4a), Inc-V4 (Fig. 2.4b), IncRes-V2 (Fig. 2.4c) and Res-101 (Fig. 2.4d) models respectively by DA-MI-FGSM attack.

by the adversarially trained model. We denote $S_{IncV3} = \{\boldsymbol{x} \in D^* | f_\theta^{IncV3}(\boldsymbol{x}) = y\}$ where the mark $IncV3$ denotes the name of the model. The ratio is formulated as follows:

$$Ratio = \frac{|S_{IncV3} \cup S_{IncV4} \cup S_{IncResV2} \cup S_{Res101} \cap S_{robust}|}{|S_{IncV3} \cup S_{IncV4} \cup S_{IncResV2} \cup S_{Res101}|} \tag{2.21}$$

where the mark $robust$ denotes the surrogate name of adversarially trained models.

From Fig. 2.9, we can see that around 90% or more 90% of the samples are correctly classified by both normally trained models and adversarially trained models. It implies that these samples are difficult to be transferred to attack the



(a) Inc-V3

(b) Inc-V4

(c) IncRes-V2

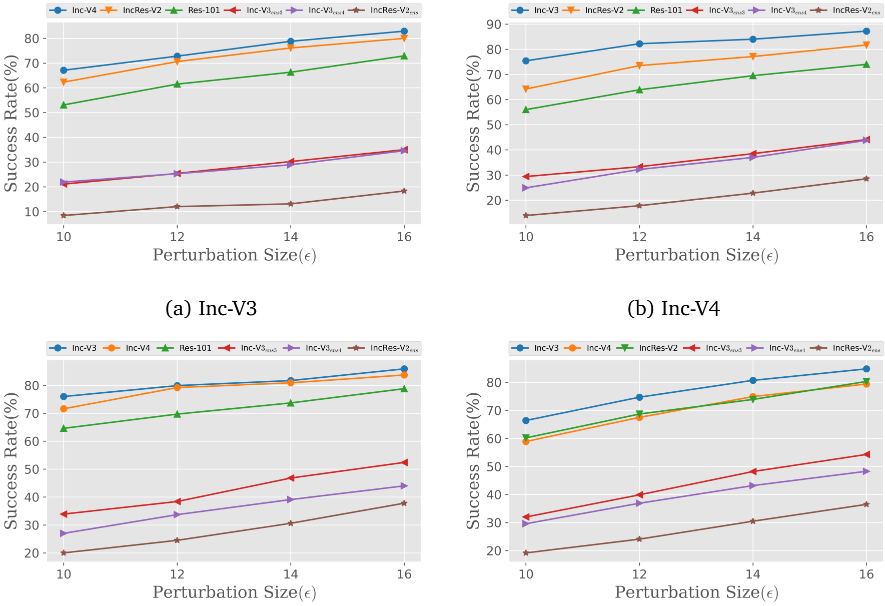
(d) Res-101

Figure 2.5: The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models when varying σ from 0 to 0.09. The adversarial examples are generated based on Inc-V3 (Fig. 2.5a), Inc-V4 (Fig. 2.5b), IncRes-V2 (Fig. 2.5c) and Res-101 (Fig. 2.5d) models respectively using DA-MI-FGSM attack.

black-box models. Therefore, the transferability of these samples improved by reducing α may not be enough to attack the black-box models successfully.

2.5 Connection of the DA-Attack to a Smoothed Classifier

Our method mitigates the overfitting problem by aggregating the attack directions of a set of examples around the input x, which is different from DIM, TIM, and SI-NI-FGSM attacks. Essentially, these methods are based on geometric transformations of the inputs, e.g. scale and translation. The successful boosting



(a) Inc-V3

(b) Inc-V4

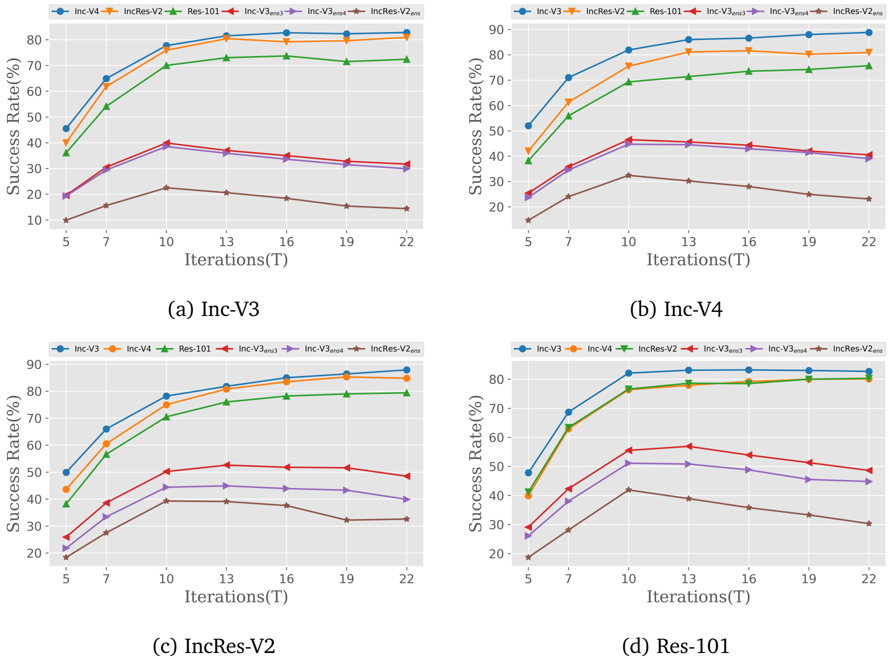
(c) IncRes-V2

(d) Res-101

Figure 2.6: The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models when varying ϵ from 10 to 16. The adversarial examples are generated based on Inc-V3 (Fig. 2.6a), Inc-V4 (Fig. 2.6b), IncRes-V2 (Fig. 2.6c) and Res-101 (Fig. 2.6d) models respectively using DA-MI-FGSM attack.

of the performance of combinations of PA-Attack with DIM or TIM (Table 2.2, Table 2.3, Table 2.4) also provides evidence that our method is orthogonal to these attacks.

For a better understanding of our method, we provide an analysis of the connection of DA-Attack to a smoothed classifier. A reasonable assumption is that adversarial examples generated by a non-smoothed classifier are more easily overfitted than that generated by a smoothed classifier. We take the Gaussian noise-smoothed classifier as an example. Formally, given a Gaussian function $g(t) = \frac{1}{\sqrt{2\pi\sigma}} \exp -\frac{t^2}{2\sigma^2}$, the Gaussian noise smoothed classifier can be presented as



(a) Inc-V3

(b) Inc-V4

(c) IncRes-V2

(d) Res-101

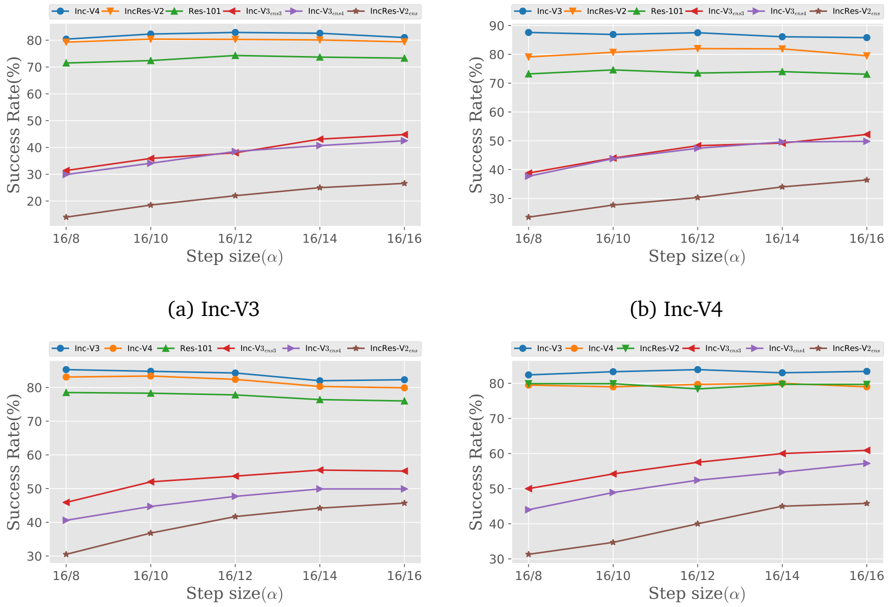
Figure 2.7: The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models when varying T from 5 to 22. The adversarial examples are generated based on Inc-V3 (Fig. 2.7a), Inc-V4 (Fig. 2.7b), IncRes-V2 (Fig. 2.7c) and Res-101 (Fig. 2.7d) models respectively using DA-MI-FGSM attack.

follows:

$$\begin{aligned}\Phi(f)(\boldsymbol{x}) &= \int_{R^n} g(\boldsymbol{y}-\boldsymbol{x}) f(\boldsymbol{y}) d\boldsymbol{y} \\ &= \mathbf{E}_{\varepsilon\in\mathcal{N}(0,\sigma^2 I)}[f(x+\varepsilon)].\end{aligned} \tag{2.22}$$

In practice, Eq. (2.22) can be empirically estimated by Monte Carlo sampling. That is, $\Phi(f)(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} f(x+\varepsilon_i), \varepsilon_i \in \mathcal{N}(0,\sigma^2 I)$. Accordingly, the gradient of $\Phi(f)(\boldsymbol{x})$ can be presented as follows:

$$\nabla_x \Phi(f)(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} \nabla_x f(x+\varepsilon_i), \varepsilon_i \in \mathcal{N}(0,\sigma^2 I). \tag{2.23}$$



(a) Inc-V3

(b) Inc-V4

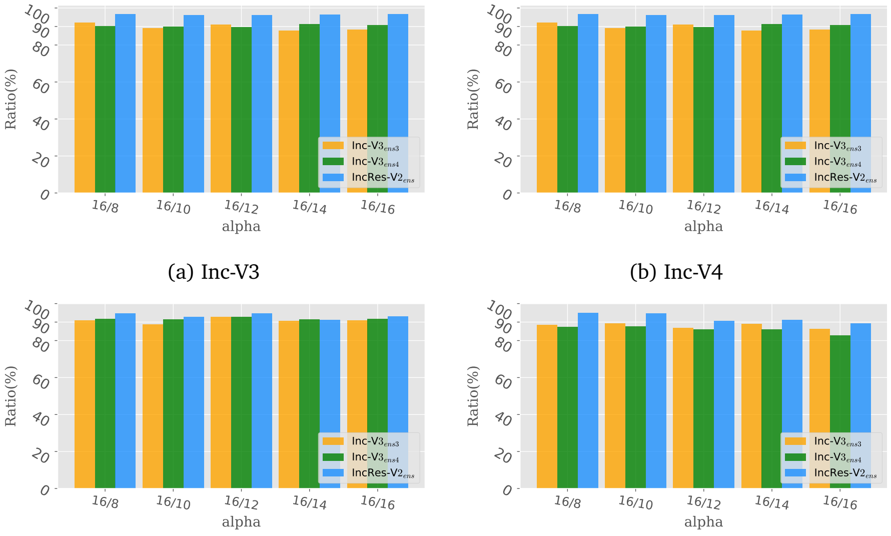
(c) IncRes-V2

(d) Res-101

Figure 2.8: The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models when varying α from $\frac{16}{8}$ to $\frac{16}{16}$. The adversarial examples are generated based on Inc-V3 (Fig. 2.8a), Inc-V4 (Fig. 2.8b), IncRes-V2 (Fig. 2.8c) and Res-101 (Fig. 2.8d) models respectively using DA-MI-FGSM attack.

Comparing Eq. (2.23) with Eq. (2.10), it can be observed that when we use the gradient instead of the projected gradient as the update direction, i.e. drop the sign function in Eq. (2.10), Eq. (2.10) will be equivalent to Eq. (2.23). $\frac{1}{N}$ can be ignored since it will not influence the attack direction. Therefore, our method will be degraded to generate adversarial examples by a smoothed classifier when we use the gradient as the attack direction directly, which also implies that DA-Attack can mitigate the overfitting issue of adversarial examples.

Actually, the smoothed classifier also could be smoothed by other noise, e.g. Uniform noise, where $g(t)$ is replaced with the uniform distribution function. Similarly, Gaussian noise is not the only choice for our DA-Attack. Uniform noise is applicable too. To provide empirical evidence for this, we conduct



(a) Inc-V3

(b) Inc-V4

(c) IncRes-V2

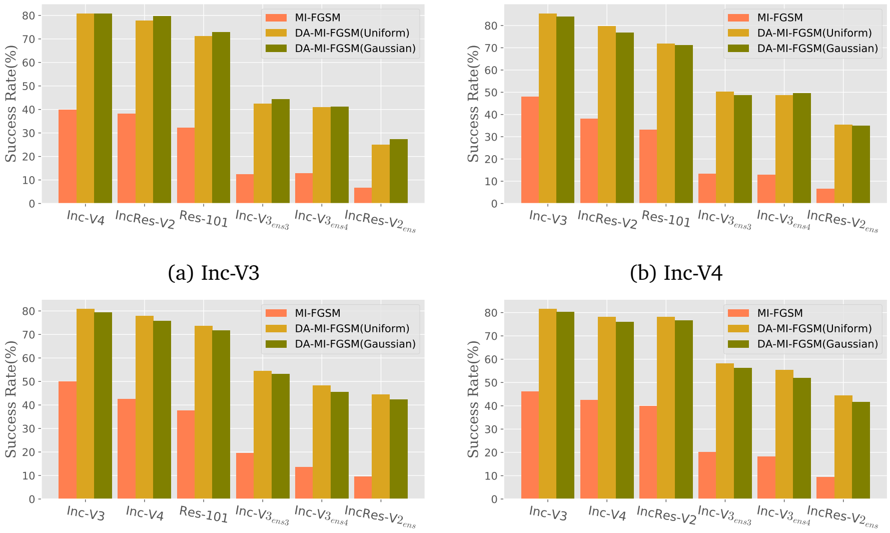
(d) Res-101

Figure 2.9: The percentage of the samples that are correctly classified by both normal models and the adversarially trained model. Adversarial examples generated by different models are shown in Fig 2.9a, Fig 2.9b, Fig 2.9c and Fig 2.9d respectively.

further experiments by replacing Gaussian noise with Uniform noise (Eq. (2.12)) sampled from $\mathbf{U}(-0.08, 0.08)$. Other hyper-parameters are set the same as in the preceding experiments (Section 2.4). The results are shown in Fig. 2.10, from which we observe that DA-Attack with Uniform noise reaches the same performance as when Gaussian noise was added. This experiment illustrates that the choice of the type of perturbations is not the key factor for DA-Attack.

2.6 Conclusion

In this study, we aimed to enhance the transferability of adversarial examples through the aggregation of attack directions in the vicinity of the input. Our proposed DA-Attack method leverages the aggregated direction. Our experiments on ImageNet, including both single-model and ensemble-based attacks, showed that our method outperforms state-of-the-art attacks. The exception was the



(a) Inc-V3

(b) Inc-V4

(c) IncRes-V2

(d) Res-101

Figure 2.10: The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models. The adversarial examples are generated based on Inc-V3 (Fig. 2.10a), Inc-V4 (Fig. 2.10b), IncRes-V2 (Fig. 2.10c) and Res-101 (Fig. 2.10d) models using DA-MI-FGSM attack with Gaussian noise and Uniform noise respectively.

IncRes-V2 model, where results were inconsistent. Our DA-Attack achieved the highest average attack success rate of 94.6% against three adversarially trained models and 94.8% against five defense models in black-box settings. These results highlight the need for stronger defense mechanisms as current defense models are not sufficient to protect against transferable adversarial attacks.

We outline several potential approaches for defending against transferable adversarial examples. The essence of existing transferable adversarial examples is that the decision boundaries of the trained models are similar. Therefore, one simple defense approach is to train ensemble models with diversified decision boundaries in order that the decision boundary of each base model is less similar to that of the white-box model. Another way is to use transferable adversarial examples as training instances, i.e. simply adding them to the training data. This idea is similar to adversarial training. The challenge here however is is to efficiently generate on-the-fly transferable adversarial examples.

Chapter 3

Calibrated Adversarial Training

Adversarial training is an approach of increasing the robustness of models to adversarial attacks by including adversarial examples in the training set. One major challenge of producing adversarial examples is to contain sufficient perturbation in the example to flip the model's output while not making severe changes in the example's semantical content. Exuberant change in the semantical content could also change the true label of the example. Adding such examples to the training set results in adverse effects. In this chapter, we present Calibrated Adversarial Training, a method that reduces the adverse effects of semantic perturbations in adversarial training. The method produces pixel-level adaptations to the perturbations based on novel calibrated robust error. We provide theoretical analysis on the calibrated robust error and derive an upper bound for it. Our empirical results show a superior performance of the Calibrated Adversarial Training over a number of public datasets.

3.1 Introduction

Despite the impressive success in multiple tasks, e.g. image classification [KH12, HZRS16], object detection [GDDM14], semantic segmentation [LSD15], deep neural networks (DNNs) are vulnerable to adversarial examples. In other words, carefully constructed small perturbations of the input can change the prediction of the model drastically [SZS+13,GSS14a]. Furthermore, these adversarial examples have shown high transferability, which greatly threatens the security of DNN models [XZZ+19,HMP+21]. This vulnerability of DNNs prohibits their adoption in applications with high risk such as autonomous driving, face recognition, and medical image diagnosis.

In response to the vulnerability of DNNs, various defense methods have been proposed. These methods can be roughly separated into two categories: 1) certified defense, and 2) empirical defense. Certified defense tries to learn provable robustness against ϵ-ball bounded perturbations [CRK19, WK18]. Empirical defense refers to heuristic methods, including augmenting training data [MMS$^+$18] (e.g. adversarial training), regularization [MDFUF18, JG18], and inspirations from biology [TKRB19]. Among all these defense methods, adversarial training has been the most commonly used defense against adversarial perturbations because of its simplicity and effectiveness [MMS$^+$18, ACW18]. Standard adversarial training takes model training as a *minmax* optimization problem (Section 3.3.2) [MMS$^+$18]. It trains a model based on on-the-fly generated adversarial examples X' bounded by uniformly ϵ-ball of input X (i.e. $\|X' - X\| \leq \epsilon$).

Although adversarial training is effective in achieving robustness, it suffers from two problems. Firstly, it achieves robustness with a severe sacrifice on natural accuracy, i.e. accuracy on natural images. Furthermore, the sacrifice will be enlarged rapidly when training with larger ϵ. Secondly, there is an underlying assumption that the on-the-fly generated adversarial examples within ϵ-ball are semantically unchanged. However, recently, Guo et al. [GFW18] and Sharma et al. [SDB19] show that adversarial examples bounded by ϵ-ball could be perceptible in some instances. Tramer et al. [TBC$^+$20] and Jacobsen et al. [JBZB19] find that there are “invariance adversarial examples” for some instances, where “invariance adversarial examples” refer to those adversarial examples that model’s prediction does not change while the true label changes. All these findings indicate that this assumption does not consistently hold, which hurts the performance of the model.

In this chapter, we first analyze the limitation of adversarial training and point out that some on-the-fly generated adversarial examples may be harmful to train models. For instance, in Figure 3.1, the adversarial examples for x_1 may be harmful since it crosses the oracle classifier’s decision boundary. To address the limitation, we propose calibrated adversarial training, which is derived from the upper bound of a new definition of robust error (Calibrated robust error). Calibrated adversarial training is composed of weighted cross-entropy loss for natural input and **KL** divergence for calibrated adversarial examples where calibrated adversarial examples are pixel-level adapted adversarial examples in order to reduce the adverse effect of adversarial examples with underlying semantic changes.

Specifically, our contributions are summarized as follows:

- Theoretically, we analyze the limitation of adversarial training and propose a new definition of robust error: Calibrated robust error. Furthermore, we derive an upper bound for the calibrated robust error.
- We propose the calibrated adversarial training based on the upper bound of calibrated robust error, which can reduce the adverse effect of adversarial examples.
- Extensive experiments demonstrate that our method achieves the best performance on both natural and robust accuracy among baselines and provides a good trade-off between natural accuracy and robust accuracy. Furthermore, it enables training with larger perturbations, which yields higher adversarial robustness.

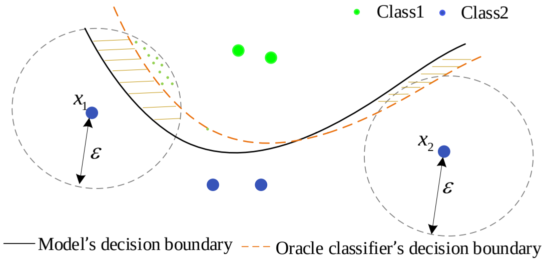

● Class1 ● Class2
x_1
ε
x_2
ε
Model's decision boundary
Oracle classifier's decision boundary

Figure 3.1: Illustration for neighborhoods of inputs and the decision boundaries.

3.2 Related Work

Many papers have proposed their variants of adversarial training for achieving either more effective adversarial robustness or a better trade-off between adversarial robustness and natural accuracy. Generally, they can be categorized into two groups. The first group adapts a loss function for outer minimization or inner maximization. For instance, Kannan et al. [KKG18] introduce a regularization term to enclose the distance between the adversarial example and the corresponding natural example. Zhang et al. [ZYJ+19b] propose a theoretically principled trade-off method (Trades). Ding et al. [DSLH19] propose Max-Margin adversarial (MMA) training by maximizing the margin of a classifier. Wang et al. [WZY+20] propose MART by introducing an explicit regularization for misclassified examples. Wu et al. [WXW20] propose Adversarial Weight Perturbation (AWP) for regularizing the weight loss landscape of adversarial training. Andriushchenko et al. [AF20] and Huang et al. [HMPP20] propose FGSM adversarial training + gradient-based regularization for achieving more effective adversarial robustness. The other group is to generate adversarial examples with adapted perturbation strength. Our work belongs to this group. Several recent works including Customized adversarial training [CLC+20], Currium adversarial training [CDLS18], Dynamic adversarial training [WMB+19], Instance adapted adversarial training [BGH19], Adversarial training with early stopping (ATES) [SCW20], Friendly adversarial training (FAT) [ZXH+20b], heuristically propose to adapt ϵ in instance-level for adversarial examples.

3.3 Preliminary

3.3.1 Notations

We denote capital letters such as X and Y to represent random variables and lower-case letters such as x and y to represent the realization of random variables. We denote by $x \in \mathcal{X}$ the sample instance, and by $y \in \mathcal{Y}$ the label, where $\mathcal{X} \in \mathbb{R}^{m \times n}$ indicates the instance space. We use $\mathcal{B}(x, \epsilon)$ to represent the neighborhood of instance x: $\{x' : \|x' - x\|_p \le \epsilon\}$. We denote a neural network classifier as $f_\theta(x)$, the cross-entropy loss as $L(\cdot)$ and Kullback-Leibler divergence as $\mathbf{KL}(\cdot||\cdot)$. We denote $P(Y|X)$ as probability output after softmax and $P(Y = y|X)$ as the probability of $Y = y$. $sgn(\cdot)$ denotes the sign function and f_{oracle} denotes the oracle classifier that maps any inputs to correct labels.

3.3.2 Standard Adversarial Training

Given a set of instances $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We assume the data are sampled from an unknown distribution $(X, Y) \sim \mathcal{D}$. The standard adversarial training can be formally expressed as follows [MMS+18]:

$$\min_{\theta} \rho(\theta), \rho(\theta) = \mathbb{E}_{(X,Y)\sim D}[\max_{X' \in \mathcal{B}(X,\epsilon)} L(f_\theta(X'), Y)]. \tag{3.1}$$

3.3.3 Projected Gradient Descent (PGD)

Madry et al. [MMS+18] utilize projected gradients to generate perturbations. Formally, with the initialization $x^0 = x$, the perturbed data in t-th step x^t can be expressed as follows:

$$x^t = \Pi_{\mathscr{B}(x,\epsilon)}(x^{t-1} + \alpha \cdot sgn(\nabla_x L(f_\theta(x^{t-1}), y))), \tag{3.2}$$

where $\Pi_{\mathscr{B}(x,\epsilon)}$ denotes projecting perturbations into the set $\mathscr{B}(x,\epsilon)$, α is the step size and $t \in \{1,2,...,T\}$. We denote PGD attack with $T = 20$ as PGD-20 and $T = 100$ as PGD-100.

3.3.4 C&W attack

Given x, C&W attack [CW17b] searches adversarial examples $\tilde{x}$ by optimizing the following objective function:

$$\|\tilde{x} - x\|_p + c \cdot h(\tilde{x}), \tag{3.3}$$

with

$$h(\tilde{x}) = \max(\max_{i \neq t} f_\theta(\tilde{x})_i - f_\theta(\tilde{x})_t, -k),$$

where $c > 0$ balances the two loss terms and k encourages adversarial examples to be classified as target t with larger confidence. This paper adopts C&W$_\infty$ attack and follows the implementation in [ZYJ$^+$19b, CDLS18] where they replace the cross-entropy loss with $h(\tilde{x})$ in PGD attack.

3.3.5 Robust Error

We introduce the definition of robust error given by [ZYJ$^+$19b, SST$^+$18].

Definition 3.3.1 (Robust Error [ZYJ$^+$19b, SST$^+$18]) *Given a set of instance x_1 ,..., $x_n \in \mathscr{X}$ and labels $y_1, ..., y_n \in \{-1, +1\}$. We assume that the data are sampled from an unknown distribution $(X, Y) \sim D$. The robust error of a classifer $f_\theta : \mathscr{X} \to \mathbf{R}$ is defined as: $\mathscr{R}_{rob}(f) := \mathbf{E}_{(X,Y)\sim D} \mathbf{1}\{\exists X' \in \mathscr{B}(X,\epsilon)\ s.t.\ f_\theta(X')Y \leq 0\}$.*

3.4 Method

3.4.1 Analysis For Adversarial Training

Current adversarial training including its variants trains a model by minimizing robust error directly, which may hurt the performance of the model. Taking standard adversarial training as an example, it first approximates robust error by inner maximization and then minimizes the approximated robust error. However, the on-the-fly adversarial examples generated by the inner maximization could be semantically damaged for some instances, e.g., in Figure 3.1, the semantical content of the adversarial examples for x_1 could be damaged since it crosses the decision boundary of f_θ. Therefore, the objective function (Eq. (3.1)) can be decomposed into two terms according to the oracle classifier's decision boundary:

$$\min_\theta \rho(\theta),\ \rho(\theta) = \mathbb{E}_{(X,Y)\sim D}[\overbrace{\max_{X'\in\mathscr{B}(X,\epsilon)} L(f_\theta(X'),Y)\mathbf{1}\{f_{oracle}(X')=Y\}}^{(a)}$$
$$+\overbrace{\max_{X'\in\mathscr{B}(X,\epsilon)} L(f_\theta(X'),Y)\mathbf{1}\{f_{oracle}(X')\neq Y\}}^{(b)}]. \tag{3.4}$$

The term (b) contributes to negative effects since the cross-entropy loss takes Y as the label of adversarial examples X' while the true label of X' is not Y. This term is equivalent to bringing noisy labels in training data, which also explains why a large perturbation magnitude in adversarial training leads to a severe drop in the natural accuracy of the model.

To overcome this limitation, we present the concept of calibrated robust error as the foundation for our defense method.

3.4.2 Calibrated Robust Error

Definition 3.4.1 (Calibrated Robust Error (Ours)) *Given a set of instances x_1 ,..., $x_n \in \mathscr{X}$ and labels $y_1,...,y_n \in \{-1,+1\}$. We assume that the data are sampled from an unknown distribution $(X,Y)\sim D$. Assume there is an oracle classifier f_{oracle} that maps any input $x\in\mathbf{R}^d$ into its true label. The calibrated robust error of a classifier $f_\theta:\mathscr{X}\to\mathbf{R}$ is defined as: $\mathscr{R}_{cali}(f) := \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X'\in\mathscr{B}(X,\epsilon)\ s.t.\ f_\theta(X')f_{oracle}(X')\leq 0\}$.*

Theorem 1 *Given a set of instance $x_1,...,x_n\in\mathscr{X}$, a classifier $f_\theta:\mathscr{X}\to\mathbf{R}$ and an oracle classifier f_{oracle} that maps any input $x\in\mathbf{R}^d$ into its true label and assumed the decision boundaries of f_θ and f_{oracle} are not overlapped* [1]*, we have:*

$$\mathscr{R}_{rob}(f)\leq\mathscr{R}_{cali}(f). \tag{3.5}$$

[1] Not overlapped denotes f_θ and f_{oracle} are not exactly the same.

Proof. We denote the set $S_R = \{(X,Y)|\forall(X,Y) \sim D, \exists X' \in \mathscr{B}(X,\epsilon)\ s.t.\ f_\theta(X')Y \leq 0\}$ and $S_{CaliR} = \{(X,Y)|\forall(X,Y) \sim D, \exists X' \in \mathscr{B}(X,\epsilon)\ s.t.\ f_\theta(X')f_{oracle}(X') \leq 0\}$.

Since $S_R \subseteq S_{CaliR} \implies \mathscr{R}_{rob}(f) \leq \mathscr{R}_{cali}(f)$, we only need to prove $S_R \subseteq S_{CaliR}$.

$$\begin{aligned}
&\forall(X,Y) \in S_R,\\
&(1)\ if\ f_\theta(X)Y \leq 0, then\ f_\theta(X)f_{oracle}(X) \leq 0 \implies X \in S_{CaliR}.\\
&(2)\ if\ \ f_\theta(X)Y > 0, then\ \exists X' \in \mathscr{B}(X,\epsilon)\ s.t\ f_\theta(X')Y \leq 0;\\
&\ \ 1)if\ f_{oracle}(X')f_\theta(X') \leq 0 \implies X \in S_{CaliR}\\
&\ \ 2)if\ f_{oracle}(X')f_\theta(X') > 0,\ then\ it\ must\ have:\\
&\ \ \ \ \exists X^{''} \in \mathscr{B}(X,\epsilon)\ s.t. f_\theta(X^{''})f_{oracle}(X^{''}) \leq 0; \qquad (3.6)
\end{aligned}$$

We prove Eq. (3.6) by the contradiction method. We assume:

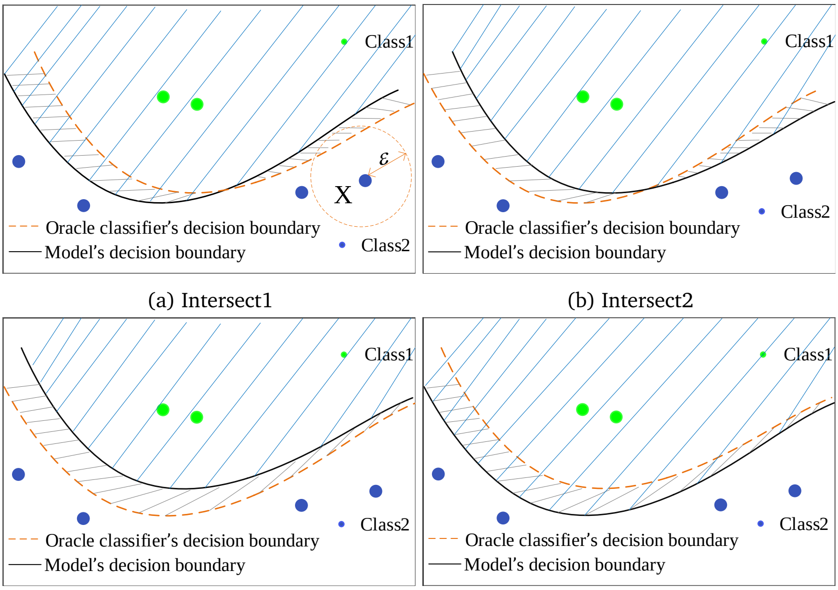
$$\forall X^{''} \in \mathscr{B}(X,\epsilon)\ s.t. f_\theta(X^{''})f_{oracle}(X^{''}) > 0\ is\ True. \qquad (3.7)$$

$$\begin{aligned}
f_\theta(X)Y > 0, f_\theta(X')Y \leq 0 &\implies the\ decision\ boundary\ of\ f_\theta\\
&\qquad crosses\ the\ \epsilon-norm\ ball\ of\ X.\\
f_\theta(X')Y \leq 0,\ f_{oracle}(X')f_\theta(X') > 0 &\implies the\ decision\ boundary\\
&\qquad of\ f_{oracle}\ crosses\ the\ \epsilon-norm\ ball\ of\ X.
\end{aligned}$$

$$\begin{aligned}
&If\ Eq.\ (3.7)\ is\ true, which\ implies\ that\ f_\theta\ and\ f_{oracle}\ have\ the\\
&same\ prediction\ on\ any\ sample\ from\ the\ \epsilon-ball\ of\ X.\\
&\implies\ the\ decision\ boundaries\ of\ f_\theta\ and\ f_{oracle}\ will\ be\\
&completely\ overlapped\ in\ \epsilon-ball\ of\ X,\ which\ contradicts\\
&the\ assumption\ of\ the\ Theorem\ 1: the\ decision\ boundaries\\
&of\ f_\theta\ and\ f_{oracle}\ are\ not\ overlapped.\\
&Therefore\ Eq.\ (3.7)\ is\ False.\\
&\implies \exists X^{''} \in \mathscr{B}(X,\epsilon)\ \ s.t. f_\theta(X^{''})f_{oracle}(X^{''}) \leq 0; Eq.\ (3.6)\ is\ proved.\\
&\implies X \in S_{CaliR}.
\end{aligned}$$

By now, we proved $\forall X \in S_R \implies X \in S_{CaliR}$. Besides, $\exists X \in S_{CaliR} \implies X \notin S_R$, e.g. the sample X in Fig. 3.2a. Therefore $S_R \subseteq S_{CaliR}$ is proved.

Besides going through the formal proof itself, we think it is useful to look into the provided visualization of the decision boundary for a more intuitive understanding. According to the spatial relationship of decision boundaries of f_θ and f_{oracle}, it can be separated into intersection and non-intersection cases (no overlap case according to the assumption in Theorem 1), which are shown in Fig. 3.2. From Fig. 3.2, for any sample (X, Y) from class 2, if $\exists X' \in \mathscr{B}(X,\epsilon)$ lies in the region filled with blue lines, it must have $\exists X^{''} \in \mathscr{B}(X,\epsilon)$ lies in the region filled with gray lines. However, if $\exists X^{'} \in \mathscr{B}(X,\epsilon)$ lies in the region filled with gray lines, it is possible that $\forall X^{'} \in \mathscr{B}(X,\epsilon)$ do not lie in the region filled with blue lines. Therefore $S_R \subseteq S_{CaliR}$.


Class1
ε
X
Oracle classifier's decision boundary
Model's decision boundary
Class2
Class1
Class2
Oracle classifier's decision boundary
Model's decision boundary
Class1
Class2
Oracle classifier's decision boundary
Model's decision boundary
Class1
Class2
Oracle classifier's decision boundary
Model's decision boundary

(a) Intersect1

(b) Intersect2

(c) Non-Intersect1

(d) Non-Intersect2

Figure 3.2: Visualization of f_θ and f_{oracle} decision boundaries. Region filled with gray lines: $\{X'|f_\theta(X')f_{oracle}(X') \leq 0\}$. Region filled with blue lines: $\{X'|f_\theta(X')Y \leq 0, Y = class2\}$.

From Theorem 1, it can be observed that minimizing robust error can be obtained by minimizing calibrated robust error.

3.4.3 Upper Bound on Calibrated Robust Error

In this section, we derive an upper bound on the calibrated robust error.

Theorem 2 (Upper Bound) *Let ψ be a nondecreasing, continuous and convex function:* $[0,1] \rightarrow [0,\infty]$. *Let* $\mathscr{R}_{\phi}(f) := \mathbf{E}\phi(f_{\theta}(X)Y)$ *and* $\mathscr{R}^{*}_{\phi} := \min_f \mathscr{R}_{\phi}(f)$, $\mathscr{R}(f) := \mathbf{E}(f_{\theta}(X)Y)$ *and* $\mathscr{R}^{*} = \min_f \mathscr{R}(f)$. *For any non-negative loss function* ϕ *such that* $\phi(0) \geq 1$, *any measurable* $f_{\theta} : \mathscr{X} \rightarrow \mathbf{R}$ *and any probability distribution on* $\mathscr{X} \times \{+1,-1\}$, *we have:*

$$\mathscr{R}_{cali}(f) - \mathscr{R}^{*} \leq \psi^{-1}(\mathscr{R}_{\phi}(f) - \mathscr{R}^{*}_{\phi}) + \mathbf{E}\Big[\max_{\substack{X' \in \mathscr{B}(X,\epsilon) \\ f_{oracle}(X')=Y}} \phi(f_{\theta}(X')Y)\Big]. \tag{3.8}$$

Proof.

$$\begin{aligned}
&\mathscr{R}_{cali}(f) - \mathscr{R}^{*} = \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathscr{B}(X,\epsilon)\ s.t.\ f_{\theta}(X')f_{oracle}(X') \leq 0\} \\
&= \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathscr{B}(X,\epsilon)\ s.t.\ f_{\theta}(X')f_{oracle}(X') \leq 0, f_{\theta}(X)Y \leq 0\} \\
&\qquad + \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathscr{B}(X,\epsilon)\ s.t.\ f_{\theta}(X')f_{oracle}(X') \leq 0, f_{\theta}(X)Y > 0\} - \mathscr{R}^{*} \\
&= \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{f_{\theta}(X)Y \leq 0\} - \mathscr{R}^{*} \\
&\qquad + \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathscr{B}(X,\epsilon)\ s.t.\ f_{\theta}(X')f_{oracle}(X') \leq 0, f_{\theta}(X)Y > 0\} \\
&\leq \psi^{-1}(\mathscr{R}_{\phi}(f) - \mathscr{R}^{*}_{\phi}) + \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathscr{B}(X,\epsilon)\ s.t.\ f_{\theta}(X')f_{oracle}(X') \leq 0, f_{\theta}(X)Y > 0\} \\
&\leq \psi^{-1}(\mathscr{R}_{\phi}(f) - \mathscr{R}^{*}_{\phi}) + \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathscr{B}(X,\epsilon)\ s.t.\ f_{\theta}(X')f_{oracle}(X') \leq 0\} \\
&\leq \psi^{-1}(\mathscr{R}_{\phi}(f) - \mathscr{R}^{*}_{\phi}) + \mathbf{E}_{(X,Y)\sim D}\max_{X' \in \mathscr{B}(X,\epsilon)}\mathbf{1}\{f_{\theta}(X')f_{oracle}(X') \leq 0\} \\
&\leq \psi^{-1}(\mathscr{R}_{\phi}(f) - \mathscr{R}^{*}_{\phi}) + \mathbf{E}_{(X,Y)\sim D}\max_{X' \in \mathscr{B}(X,\epsilon)}\phi(f_{\theta}(X')f_{oracle}(X'))
\end{aligned}$$

$Let\ f_{oracle}(X') = Y,\ then,$

$$\mathscr{R}_{cali}(f) - \mathscr{R}^{*} \leq \psi^{-1}(\mathscr{R}_{\phi}(f) - \mathscr{R}^{*}_{\phi}) + \mathbf{E}_{(X,Y)\sim D}\max_{\substack{X' \in \mathscr{B}(X,\epsilon) \\ f_{oracle}(X')=Y}} \phi(f_{\theta}(X')Y)$$

The first inequality holds when ϕ is a classification-calibrated loss [ZYJ+19b, BJM06]. Classification-calibrated loss contains the cross-entropy loss, hinge loss, **KL** divergence and etc.

From the upper bound, it can be observed:

- If the oracle classifier's decision boundary crosses ϵ-ball, the upper bound is decided by the adversarial examples that are close to the oracle classifier's decision boundary. If the oracle classifier's decision boundary does not cross ϵ-ball, the upper bound is decided by the adversarial examples that are close to the boundary of ϵ-ball.

- Minimizing $\mathscr{R}_{\phi}(f) + \mathbf{E}\Big[\max_{\substack{X' \in \mathscr{B}(X,\epsilon) \\ f_{oracle}(X')=Y}} \phi(f(X')Y)\Big]$ can reduce the calibrated robust error. From Theorem 1, we can know that calibrated robust error is the upper bound of robust error. Therefore, it also reduces the robust error of the model.

3.4.4 Method for Defense

From the upper bound, we define the general objective function as follows:

$$\min_{\theta} \mathbf{E}\Big[\phi(f_{\theta}(X)Y) + \max_{\substack{X' \in \mathscr{B}(X,\epsilon) \\ f_{oracle}(X')=Y}} \phi(f_{\theta}(X')Y)\Big]. \tag{3.9}$$

The first term in Eq. (3.9) is the surrogate loss of misclassification on natural data, and we design it as a cross-entropy weighted by $(1 - P(Y = y|X))$ where $P(Y = y|X)$ represents the probability of target label. Formally, it is expressed as:

$$\phi(f_{\theta}(X)Y) = L(f_{\theta}(X), Y) \cdot (1 - P(Y = y|X)). \tag{3.10}$$

The second term in Eq. (3.9) is the surrogate loss on adversarial examples. However, it can not be solved directly since f_{oracle} is unknown. Therefore, we propose an approximate solution with two steps. Firstly, we generate adversarial examples based on $\max_{X' \in \mathscr{B}(X,\epsilon)} \phi(f_{\theta}(X')Y)$. Secondly, we adapt the adversarial examples at pixel-level such that it approximately satisfies the constraint $f_{oracle}(X') = Y$ and we name the pixel-level adapted adversarial examples as **calibrated adversarial examples**. We rewrite $\max_{\substack{X' \in \mathscr{B}(X,\epsilon) \\ f_{oracle}(X')=Y}} \phi(f_{\theta}(X'), Y)$ as follows:

$$X' = argmax_{X' \in \mathscr{B}(X,\epsilon)} \phi(f_{\theta}(X')Y) \tag{3.11}$$

$$X'_{cali} = X + M \odot (X' - X),\ M \in \mathbb{R}^{m \times n}, M[i,j] \in (0,1), \tag{3.12}$$

where the $\odot$ denotes Hadamard product. From Eq. (3.11) and Eq. (3.12), we can see that calibrated adversarial examples X'_{cali} are obtained by adapting adversarial perturbations with soft mask M. Please refer to Section 3.5.2 for a better understanding of how the mask M adapts the adversarial perturbations.

X' can be solved by various adversarial attacks, e.g., the PGD attack. Therefore, the problem of the inner maximization in Eq. (3.9) is transformed to find a proper soft mask M. Considering that soft mask M relies on inputs X and the perturbations $\delta = X' - X$, we propose to learn it by a neural network g_φ, which is defined as follows:

$$M = g_\varphi(X, \delta). \tag{3.13}$$

Therefore, by replacing $\phi(f_\theta(X)Y)$ with Eq. (3.10) and X' with X'_{cali}, the objective function (Eq. (3.9)) is transformed to follows:

$$\min_\theta \mathbf{E}_{(X,Y)\sim D}[L(f_\theta(X), Y) \cdot (1 - P(Y = y|X)) + \beta \cdot \phi(f_\theta(X'_{cali})Y)], \tag{3.14}$$

where X'_{cali} is solved by Eq. (3.12), and β is a hyper-parameter for balancing two terms. In practice, we follow [ZYJ+19b, WZY+20] to use **KL** divergence for the surrogate loss $\phi(\cdot)$ in the outer minimization step. Thus, Eq. (3.14) can be reformulated as follows:

$$\min_\theta \mathbf{E}_{(X,Y)\sim D}[L(f_\theta(X), Y) \cdot (1 - P(Y = y|X)) + \beta \cdot \mathbf{KL}(P(Y|X'_{cali})||P(Y|X))]. \tag{3.15}$$

From Eq. (3.15), it can be observed that there are two main differences with other variants of adversarial training, e.g., AT, Trades, MART, etc.

- We use weighted cross-entropy loss instead of cross-entropy loss in order to make the loss function pay more attention to misclassified samples.
- The **KL** divergence is based on calibrated adversarial examples that reduce the adverse of some adversarial examples because calibrated adversarial examples are expected to be satisfied with $f_{oracle}(X'_{cali}) = Y$.

Finally, we design the objective function for $g_\varphi(X, \delta)$ based on the two constraints: (1) X'_{cali} should be close to X' as far as possible in order to keep the inner maximization constraint in Eq. (3.9). (2) X'_{cali} is expected to be satisfied with $f_{oracle}(X'_{cali}) = Y$. Therefore, the objective function for $g_\varphi(X, \delta)$ is designed as follows:

$$\min_\varphi \mathbf{E}_{(X,Y)\sim D}[\mathbf{KL}(P(Y|X'_{cali})||P(Y|X')) + \beta_1 \cdot L(f_\theta(X'_{cali}), Y)], \tag{3.16}$$

where **KL** divergence term corresponds to the constraint (1) and cross-entropy loss $L(\cdot)$ corresponds to the constraint (2). β_1 is the hyper-parameter that controls the strength of the constraint (2).

We denote our method as calibrated adversarial training with PGD attack (CAT$_{cent}$) if X' is solved by PGD attack, calibrated adversarial training with C&W$_\infty$ attack (CAT$_{cw}$) if X' is solved by C&W$_\infty$ attack.

3.5 Experiments

In this section, we first conduct extensive experiments to assess the effectiveness of our approach in achieving natural accuracy and adversarial robustness, then we conduct experiments for understanding the proposed method.

3.5.1 Evaluation on Robustness and Natural Accuracy

Experimental settings

Two datasets are used in our experiments: MNIST [LeC98], and CIFAR-10 [KNH10]. For MNIST, all defense models are built on four convolution layers and two linear layers. For CIFAR-10, we use PreAct ResNet-18 [HZRS16] and WideResNet-34-10 [ZK16] models. Following previous studies [ZYJ+19b,WXW20], Robustness is measured by robust accuracy against white-box and black-box attacks. For white-box attack, we adopt PGD-20/100 attack [MMS+18], FGSM attack [GSS14a] and C&W$_\infty$ [CW17b]. For black-box attacks, we adopt the query-based attack: Square attack [ACFH20].

Baselines. Standard adversarial training and the three latest defense methods are considered: 1)TRADES [ZYJ+19b], 2)MART [WZY+20], 3)FAT [ZXH+20b].

Hyper-parameter settings. During training phase, for MNIST, we set $T = 20$, $\epsilon = 0.3$, $\alpha = \epsilon/T$ for the training attack, and set $\beta = 1$, $\beta_1 = 0.3$ by default. For CIFAR-10, we set $T = 10$, $\alpha = 2/255$, $\epsilon = 8/255$ for the training attack and set $\beta = 5$ by default. We train models with $\beta_1 = 0.05, 0.1, 0.3$ respectively. For all baselines, they are trained using the official code that their authors provided, and the hyper-parameters for them are set as per their original papers.

During the test phase, for MNIST, we set $\epsilon = 0.3$ and $\alpha = 0.015$ for the PGD attack. For CIFAR-10, we set $\epsilon = 8/255$ and $\alpha = 0.003$ for PGD attack. And we follow the implementation in [ZXH+20b] for C&W$_\infty$ attack where $\epsilon = 0.031$, $\alpha = 0.003$, $T = 30$ and $k = 50$.

Note that during the training process, we use the PGD attack with a random start, i.e. adding random perturbation of $[-\epsilon, \epsilon]$ to the input before PGD perturbation. But for the test in our experiments, we use PGD attack without random

Table 3.1: Evaluation on MNIST. The value beside the model name denotes the max perturbation magnitude used in the training phase. -: denotes the training loss fails in decrease. We report the mean value with 5 repeated runs and skip the standard deviations since they are small ($< 0.4\%$), which hardly affects the results.

Models	Natural	PGD-20	PGD-100
AT(0.3)	99.2	93.4	92.3
AT(0.4)	-	-	-
TRADES(0.3)*	99.3	94.9	92.9
TRADES(0.4)*	99.1	95.3	91.6
CAT$_{cent}$(0.3)	**99.3**	95.4	93.2
CAT$_{cent}$(0.4)	99.2	96.8	95.8
CAT$_{cw}$(0.3)	99.1	96.2	95.0
CAT$_{cw}$(0.4)	99.1	**97.1**	**96.2**

* Model is trained with $\beta = 1.0$.

start by default [2].

Evaluation on White-box Robustness

This section shows the evaluation against white-box attacks. All attacks have full access to model parameters. We first conduct an evaluation on a simple benchmark dataset: MNIST and then conduct an evaluation on a complex dataset: CIFAR-10.

MNIST. Table 3.1 reports natural accuracy and robust accuracy under PGD-20 and PGD-100 respectively. For baselines, we do not include results from FAT and MART since they do not provide training codes for MNIST. From Table 3.1, we can see that the proposed method can achieve higher natural accuracy and robust accuracy compared with standard adversarial training. Besides, we notice that with larger $\epsilon = 0.4$, adversarial robustness can be boosted further by our defense method.

CIFAR-10. We evaluate the performance based on two benchmark architectures, i.e., PreAct ResNet-18 and WideResNet-34-10. All defense models are tested under the same attack settings as described in Section 3.5.1 except for *FAT* on WideResNet-34-10 since this evaluation is copied from their paper directly

[2] We find that PGD attack (restart=1) without random start is stronger than that with random start.

Table 3.2: Evaluation on CIFAR-10 for PreAct ResNet-18 under white-box setting.

Models	Natural	FGSM	PGD-20	PGD-100	CW$_\infty$	Avg
AT	83.0	57.3	52.9	51.9	50.9	59.2
TRADES(β:6)	82.8	57.6	52.8	51.7	50.9	59.2
MART (λ:5)	83.0	**60.2**	53.9	52.3	49.9	59.9
FAT(β:6)	85.1	58.3	52.1	50.5	50.4	59.3
CAT$_{cent}$(β_1 : 0.05)	84.1 ± 0.3	59.5 ± 0.2	**55.6 ± 0.3**	**54.9±0.3**	50.8±0.2	**61.0**
CAT$_{cent}$(β_1 : 0.1)	85.9 ±0.2	58.5±0.3	54.1 ±0.1	53.4 ±0.06	50.44±0.3	60.4
CAT$_{cw}$(β_1 : 0.05)	84.2 ±0.3	58.9±0.2	55.3 ±0.4	54.5 ±0.5	**51.3±0.3**	60.9
CAT$_{cw}$(β_1 : 0.1)	85.1 ±0.5	58.9±0.3	54.9 ±0.5	54.1 ±0.4	51.2±0.1	60.8
CAT$_{cent}$(β_1 : 0.3)	88.0 ±0.2	57.0±0.4	51.1 ±0.5	49.9 ±0.4	47.8±0.2	58.8
CAT$_{cw}$(β_1 : 0.3)	**88.1** ±0.1	57.4±0.5	51.5 ±0.1	50.1 ±0.2	48.8±0.2	59.2

where it is evaluated with $\epsilon = 0.031$ for PGD attack. Table 3.2 and Table 3.3 report natural accuracy and robust accuracy on the test set. "Avg" denotes the average of natural accuracy and all robust accuracy, and it indicates the overall performance on both natural accuracy and robust accuracy. For our method, we report the mean + standard deviation with 5 repeated runs.

From Table 3.2 and Table 3.3, it can be seen that our method achieves the best performance on both natural accuracy and robust accuracy under all attacks except for FGSM among baselines. Moreover, with $\beta_1 = 0.3$, our method improves natural accuracy with a large margin while keeping comparable performance with baselines on robust accuracy. Besides, our method achieves a high "Avg" value, which indicates our method has a good trade-off between natural accuracy and robust accuracy. Finally, we observe that the robustness achieved by our method has smaller accuracy under stronger attacks, i.e. PGD-100 and CW$_\infty$, than weaker attacks, i.e. FGSM and PGD-20. It indicates that the robustness achieved by our method is not caused by "gradient masking" [ACW18].

Evaluation on Black-box Robustness

We conduct evaluations on black-box settings. We choose to use Square attack [ACFH20] in our experiments. The square attack is a black-box attack that is efficient in terms of query use and has been shown to achieve performance comparable to white-box attacks and resist "gradient masking" [ACFH20]. In our experiments, we set hyper-parameters $n_{queries} = 5000$ and $eps = 8/255$ for the square attack. The experiments are carried out on CIFAR-10 test set based on PreAct ResNet-18 and WideResNet-34-10 architectures. Results are shown in Table 3.4. It can be seen that our method achieves the best accuracy among all

Table 3.3: Evaluation on CIFAR-10 for WideResNet-34-10 under white-box setting.

Models	Natural	FGSM	PGD-20	PGD-100	CW_∞	Avg
AT	86.1	61.8	56.1	55.8	54.2	62.8
TRADES(β:6)	84.9	60.9	56.2	55.1	54.5	62.3
MART (λ:5)	83.6	61.6	57.2	56.1	53.7	62.5
FAT(β:6)	86.6±0.6	61.9±0.6	55.9±0.2	55.4±0.3	54.3±0.2	62.8
$\text{CAT}_{cent}(\beta_1:0.05)$	86.6±0.1	60.9 ± 0.1	57.7 ± 0.1	57.2 ±0.2	53.9 ±0.6	63.3
$\text{CAT}_{cent}(\beta_1:0.1)$	87.5±0.51	61.5 ±0.5	57.2 ±0.3	56.6 ±0.4	54.0±0.4	63.4
$\text{CAT}_{cw}(\beta_1:0.05)$	86.4±0.1	**62.7 ±0.2**	**59.7 ±0.1**	**58.7 ±0.3**	**56.0±0.1**	**64.7**
$\text{CAT}_{cw}(\beta_1:0.1)$	87.4±0.1	62.3 ±0.1	58.6 ±0.2	57.3 ±0.19	55.6±0.07	64.2
$\text{CAT}_{cent}(\beta_1:0.3)$	88.9 ± 0.4	59.8 ±0.6	54.8 ±0.7	53.9 ±0.6	51.6±0.2	61.8
$\text{CAT}_{cw}(\beta_1:0.3)$	**89.3±0.1**	60.8±0.27	55.1±0.3	53.2±0.5	52.6±0.4	62.2

Table 3.4: Evaluation on CIFAR-10 for PreAct ResNet-18 and WideResNet-34-10 under black-box setting. -: Not Available.

Models	ResNet	WRN
AT	55.12	59.19
TRADES	54.85	59.0
MART	54.98	57.7
FAT	55.35	-
$\text{CAT}_{cent}(\beta_1:0.05)$	56.4±0.1	59.1±0.5
$\text{CAT}_{cent}(\beta_1:0.1)$	56.4±0.1	59.6±0.8
$\text{CAT}_{cw}(\beta_1:0.05)$	56.3 ±0.2	60.9±0.1
$\text{CAT}_{cw}(\beta_1:0.1)$	**56.5 ±0.1**	**60.9±0.2**

baselines under square attack. Besides, by comparing Table 3.4 with Table 3.2 and Table 3.3, we can find that accuracy under black-box attack is lower than under white-box attack like PGD and CW_∞ attacks. It demonstrates that adversarial robustness achieved by our method is not due to “gradient masking ” [ACW18].

3.5.2 Understanding the Proposed Defense Method

Visualization of Soft Mask M

We visualize the learned soft mask M for further understanding the calibrated adversarial examples. As shown in Figure 3.3, natural images are randomly selected from MNIST, and adversarial examples are generated by PGD-20 attack

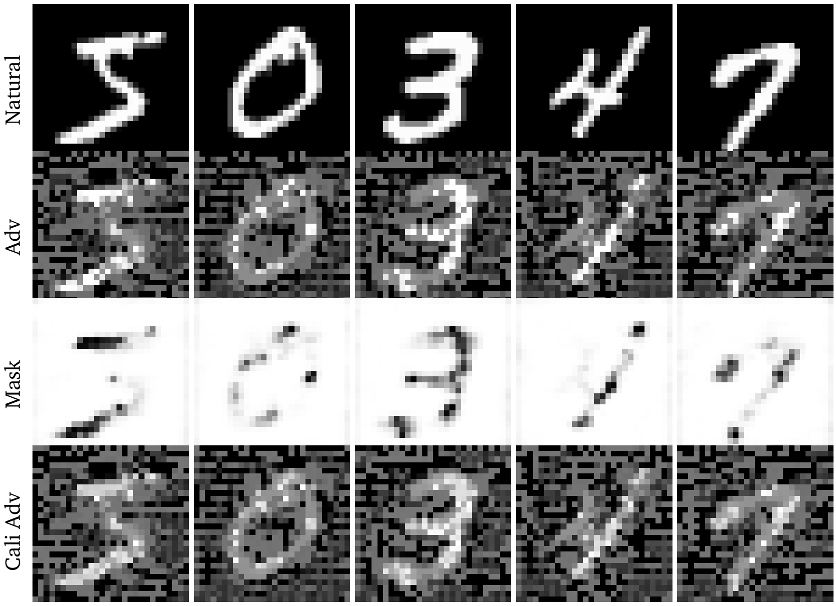
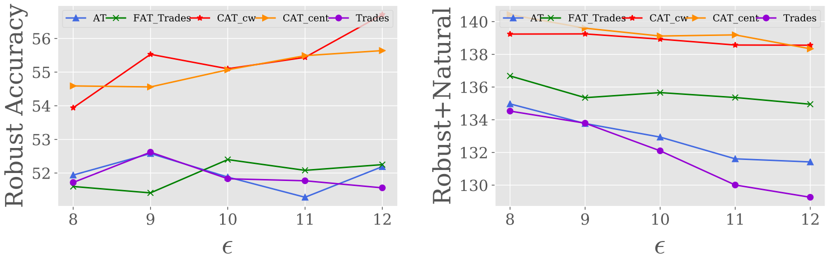


Figure 3.3: Visualization of soft mask M.

with $\epsilon = 0.4$. Soft masks and calibrated adversarial examples are generated accordingly. It can be observed that soft masks have high values on the background but low values on the digit, indicating an attempt to minimize perturbations on the digit. Furthermore, by comparing calibrated adversarial examples with regular adversarial examples, we find that pixel values on digits for calibrated adversarial examples tend to be homogeneous, which is more consistent with them on natural images. This suggests that soft masks aim to preserve semantic information and prevent adversarial examples from altering it, which can impact model performance.

Training with Larger Perturbation Bound

Our method adapts adversarial examples for mitigating the adverse effect, which enables a model trained with larger perturbations. To verify the performance, we conduct experiments on PreAct ResNet-18 models trained with $\epsilon = 8, 9, 10, 11, 12$ respectively and test them on CIFAR-10 test set. Baselines are trained with their official codes. Results are shown in Figure 3.4. From Figure 3.4a, it can be observed that our method has a clearly increasing trend on robust accuracy with the increase of ϵ. From Figure 3.4b, we can see that the sum of robust accuracy and natural accuracy has a slightly decreasing trend for our method, indicating a trade-off between robust accuracy and natural accuracy. However, our method's descending grade is lower than Trades and AT, which also verifies that our method has a good trade-off between robust accuracy and natural accuracy.



(a) Robust

(b) Robust+Natural

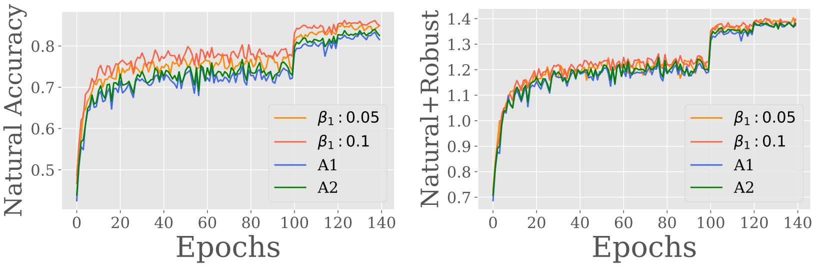
Figure 3.4: Evaluation on models trained with larger ϵ. Robust accuracy is calculated by PGD-100 attack without random start.β_1 is fixed to 0.1 for CAT_{cw} and CAT_{cent}.

Ablation Study

We empirically verify the effect of weighted cross-entropy loss and soft mask M. Besides, we compare the effect of different loss functions selected in Eq. (3.11) for generating adversarial examples.

Effect of the weighted cross-entropy loss and mask M. We remove M by replacing X'_{cali} with X' and remove $L(f_\theta(X), Y) \cdot (1 - P(Y = y|X))$ by replacing it with $L(f_\theta(X), Y)$. We train PreAct ResNet-18 models based on CAT$_{cent}$ by removing both weighted cross-entropy loss and M (marked as A1 model), and by removing M only (marked as A2 model). We plot natural accuracy and robust accuracy on the CIFAR-10 test set. Robust accuracy is computed by PGD-10 with random start ($\alpha = 2/255, \epsilon = 8/255$). Results are reported in Figure 3.5. It can be observed that after removing soft mask M, there is a clearly decrease in natural accuracy and overall performance (natural+robust accuracy). Furthermore, after removing weighted cross-entropy loss, there is a slight decrease in natural accuracy.

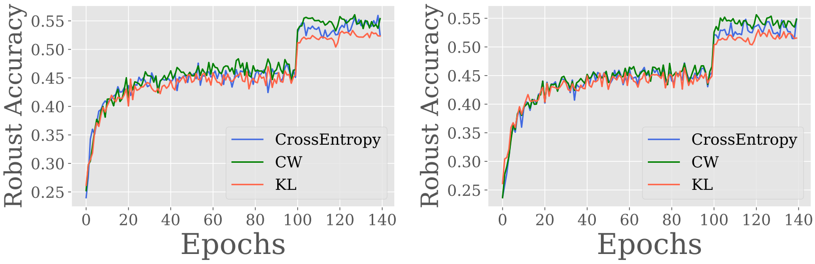
Comparison of different loss functions. There are many choices for the surrogate loss in Eq. (3.11) used to generate adversarial examples, e.g., cross-entropy loss, KL divergence used in Trades [ZYJ+19b], CW_∞ loss. Here we evaluate the effect of these three losses in our method. We plot robust accuracy on CIFAR-10 test set for $\beta_1 = 0.1$ and $\beta_1 = 0.05$ respectively, and robust accuracy is calculated by PGD-10 attack with random start ($\alpha = 2/255, \epsilon = 8/255$). The experiments are based on PreAct ResNet-18 model. Results are shown in Figure 3.6 and it can be seen that **KL** divergence is less effective in achieving robustness than cross-entropy loss and CW_∞ loss for both $\beta_1 = 0.1$ and $\beta_1 = 0.05$ settings.



(a) Natural

(b) Natural+Robust

Figure 3.5: The ablation Experiments. A1: Model trained by CAT_{cent} with removing both soft mask M and $(1 - P(Y = y|X))$. A2: Model trained by CAT_{cent} with removing soft mask M only. $\beta_1 : 0.1, 0.05$ denote models trained by CAT_{cent} with setting $\beta_1 = 0.1, 0.05$ respectively.



(a) $\beta_1 = 0.05$

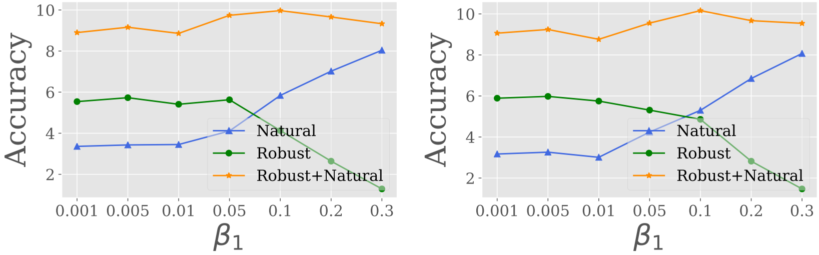
(b) $\beta_1 = 0.1$

Figure 3.6: Comparison of different loss functions on achieving adversarial robustness.

Analysis for Hyper-parameter β_1

There are two hyper-parameters, β and β_1, in our method. β has the same effect as λ in MART [WZY+20] and Trades [ZYJ+19b]. It controls the strength of the regularization for robustness. β_1 controls the strength that pushes calibrated adversarial examples to be the same class as the input X. In this section, we mainly show the effect of β_1 on robust accuracy and natural accuracy. We train models with β_1 varying from 0.001 to 0.3 based on PreAct ResNet-18 architecture. The robust accuracy is calculated on the CIFAR-10 test set by PGD-20 attack without random start.

The trends are shown in Figure 3.7. From Figure 3.7, it can be observed that when increasing the value of β_1, natural accuracy has remarkable growth. Meanwhile, PGD+Natural accuracy increases when β_1 is from 0.01 to 0.1, which implies that calibrated adversarial examples release the negative effect of adversarial examples to some degree. With continuously increasing β_1, there is a large drop in robust accuracy. It is because a large β_1 will reduce adversarial perturbation strength. However, it can be observed that there is a good trade-off for large β_1 between natural accuracy and robust accuracy. For example, with $\beta_1 = 0.3$, CAT$_{cw}$ achieves 88.08 ± 0.07 for natural accuracy while keeping 51.46 ± 0.11 for robust accuracy, which is much better than the trade-off achieved by Trades [ZYJ+19b] where natural accuracy is 87.91 and robust accuracy is 41.50 [3].



(a) CAT$_{cent}$ (b) CAT$_{cw}$

Figure 3.7: Impact of hyper-parameter β_1 on the performance of natural accuracy and robust accuracy. Note: The natural accuracy shown in the figure is $(natural\ accuracy - 80)$ and the robust accuracy shown in the figure is $(robust\ accuracy - 50)$.

[3] Results are copied from [ZYJ+19b]

3.6 Conclusion

In this chapter, we proposed a new definition of robust error, i.e. calibrated robust error for adversarial training. We derived an upper bound for it and enabled a more effective way of adversarial training that we call calibrated adversarial training. The results of our extensive experiments show that calibrated adversarial training significantly improves natural accuracy while maintaining strong robust accuracy, making it a leading approach among state-of-the-art methods. Our method also strikes the best balance between natural accuracy and robust accuracy among baselines.

Chapter 4

In-Time Refining Optimization Trajectories Toward Improved Robust Generalization

Despite the fact that adversarial training has become the de facto method for improving the robustness of deep neural networks, it is well-known that vanilla adversarial training suffers from daunting robust overfitting, resulting in unsatisfactory robust generalization. A number of approaches have been proposed to address these drawbacks such as extra regularization, adversarial weights perturbation, and training with more data over the last few years. However, the robust generalization improvement is yet far from satisfactory. In this chapter, we approach this challenge with a brand new perspective – refining historical optimization trajectories. We propose a new method named **Weighted Optimization Trajectories (WOT)** that leverages the optimization trajectories of adversarial training in time. We have conducted extensive experiments to demonstrate the effectiveness of WOT under various state-of-the-art adversarial attacks. Our results show that WOT integrates seamlessly with the existing adversarial training methods and consistently overcomes robust overfitting, resulting in better adversarial robustness. For example, WOT boosts the robust accuracy of AT-PGD under AA-L_∞ attack by 1.53% ~ 6.11% and meanwhile increases the clean accuracy by 0.55%~5.47% across the SVHN, CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets.

4.1 Introduction

Deep neural networks (DNNs) have achieved enormous breakthroughs in various fields, e.g., image classification [HDY+12,HZRS16], speech recognition [HDY+12], object detection [GDDM14] and etc. However, it has been shown that they are vulnerable to adversarial examples, i.e., carefully crafted imperceptible perturbations on inputs can easily change the prediction of the model [SZS+13, GSS14a]. The vulnerability of DNNs hinders their applications in risk-sensitive tasks such as face recognition, autonomous driving, and medical diagnostics. While various methods have been proposed to obtain robustness against adversarial perturbations, adversarial training [MMS+17] is the leading approach to achieve adversarial robustness.

However, the vanilla adversarial training usually suffers from daunting robust overfitting, resulting in poor robust generalization[1] [RWK20]. To tackle this issue, a number of methods from different perspectives have been proposed including but not limited to training with more data [SST+18, RGC+21, SMH+21, CRS+19, AUH+19], adversarial weights perturbation [WXW20, YHG+21], and knowledge distillation and stochastic weights averaging (SWA) [CZL+20]. Recently, Stutz et al. [SHS21] empirically show that the improved adversarial robustness can be attributed to the flatter loss landscape at the minima.

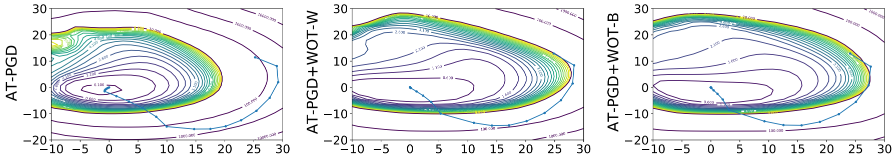


Figure 4.1: Visualization of loss contours and optimization trajectories for AT-PGD and AT-PGD+WOT-W/B (Ours). The experiments are conducted on CIFAR-10 with PreRN-18.

Although the generalization properties of SGD-based optimizer under standard training setting have been well studied [ZLR+17, EEPK05, ZLZ18, HRS16], the corresponding robust generalization property under adversarial setting has not been fully explored. Among previous studies, Chen et al. [CZL+20] heuristically adopts stochastic weight averaging (SWA) and average model weights along the optimization trajectory, which potentially mitigates robust overfitting.

[1] Robust generalization refers to the difference between the model's performance on adversarial examples in the training set and the test set, following previous work [CZL+20, WXW20, SHS21].

However, it has been shown that naive weight averaging is not general enough to fundamentally address this problem, still prone to robust overfitting [RGC+21]. Instead of simply averaging weights, we propose a new approach - **Weighted Optimization Trajectories** (briefly **WOT)** for the first time showing that we can largely improve the flatness of solutions of existing adversarial training variants by periodically refining a set of historical optimization trajectories. Compared with the existing approaches, our method has three unique design contributions: ❶ our refinement is obtained by maximizing the robust accuracy on the *unseen hold-out set*, which is naturally advantageous to address the overfitting issue; ❷ our refinement is performed on a set of previous optimization trajectories rather than solely on previous weights; ❸ we further propose a block-wise trajectory refinement, which significantly enlarges the optimization space of refinement, leading to better robust performance. We conduct rigorous experiments to demonstrate the effectiveness of these design novelties in Section 4.4.1 as well as the ablation study in Section 4.4.3. Simple as it looks in Figure 4.1, the optimization trajectories after refining converge to a flatter loss valley compared to the vanilla AT-PGD, indicating the improved robust generalization [WXW20, WWX20a, SHS21].

Extensive experiments on different architectures and datasets show that WOT seamlessly mingles with the existing adversarial training methods with consistent robust accuracy improvement. For example, WOT-B directly boosts the robust accuracy over AT-PGD (early stops) under AA-L_∞ attack by 6.11%, 1.53%, 1.57%, and 4.38% on SVHN, CIFAR-10, CIFAR-100, and Tiny ImageNet, respectively; meanwhile improves the corresponding clean accuracy by 0.55% ~ 5.47%. Moreover, we show that WOT can completely prevent robust overfitting across different attack approaches, including the strongest one off-the-shelf - AA-L_∞ attack.

4.2 Related Work

Robust Overfitting and its Mitigation. Recently, Rice, Wong, and Kolter [RWK20] identified robust overfitting in AT that robust accuracy in test set degrades severely after the first learning rate decay and found that early stop is an effective strategy for mitigating robust overfitting. Following [RWK20], several studies have been proposed to explain and mitigate the issue of robust overfitting [WXW20, SSFJ21, CZL+20, DXY+21, CZW+22, SHS21]. Chen et al. [CZL+20] showed that stochastic weight average (SWA) and knowledge distillation can mitigate the issue of robust overfitting decently and Singla et al. [SSFJ21] found that low curvature activation helps to mitigate robust overfitting problem. Dong et al. [DXY+21] took a step further to explain that robust overfitting may be caused by the memorization of hard samples in the final phase of training. [WXW20,YHG+21,SHS21] demonstrated that a flattened loss landscape improves robust generalization and reduces robust overfitting problem, which is in line with the sharpness studies in standard training setting [FKMN20,JNM+19,DR17]. Among these studies, SWA is technically close to our method. In essence, SWA does a post-process for optimization trajectories heuristically while our method refines optimization trajectories with respect to the robust performance on an unseen dataset. Furthermore, our method does the refinement in time instead of a post-process.

4.3 Methodology

In this section, we introduce **weighted optimization trajectories (WOT)**, a carefully designed method that refines the optimization trajectory of adversarial training towards a flatter region in the training loss landscape, to avoid robust overfitting. Specifically, WOT collects a set of historical optimization trajectories and further learns a weighted combination of them explicitly on the unseen set. The sketch map of WOT is shown in Figure 4.2. Concretely, WOT contains two steps: (1) collect optimization trajectories of adversarial training. (2) re-weight collected optimization trajectories and optimize weights according to the robust loss on an unseen set. Two unanswered problems of this process are how to collect optimization trajectories and how to construct the objective function of optimizing weights. We give detailed solutions as follows.

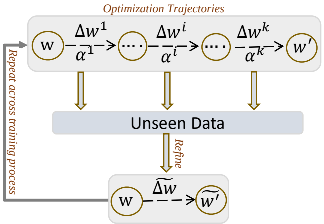


Figure 4.2: Sketch map of WOT.

4.3.1 WOT: Optimization Trajectories

We denote optimization trajectories as the consecutively series status of weights in weight space after n steps optimization. Formally, given a deep neural network f with the parameter $w \in \mathcal{W}$. n steps optimization trajectories of adversarial training is denoted as $\{w^1, w^2 ... w^i, ..., w^n\}$ where w^i is the weight after $i-$th optimization. This process can also be simplified as follows: $\{w^1, \Delta w^1 ... \Delta w^i, ..., \Delta w^{n-1}\}$

where $\Delta w^i = w^{i+1} - w^i$. In practice, it is time-consuming and space-consuming to collect the weights of each batch optimization step and it is also not necessary to collect the weights at so high frequency (See details in Figure 4.5). Therefore, we propose to collect weights for every m batch optimization step and the collected trajectories with n optimization steps are re-denoted as follows:

$$\Delta Ws = \{w^1, \Delta w^1, ..., \Delta w^i, ..., \Delta w^k\}, \tag{4.1}$$

where $k = \frac{n}{m}$. For brevity, we call m as Gaps that controls the length between two consecutively collections and k as the number of Gaps that controls the number of weights that are collected.

4.3.2 WOT: Objective Function

We design the objective function based on historical optimization trajectories of model training. From the description of optimization trajectories introduced above, the weights w' with n batch optimization steps from w can be written as $w' = w + \Delta w^1 + ... + \Delta w^i + .. + \Delta w^k$. Since WOT refines the optimization trajectories by re-weighting them, the weights $\widetilde{w'}$ after refining optimization trajectories can be expressed as follows:

$$\widetilde{w'} = w + \widetilde{\Delta w}, \ \widetilde{\Delta w} = \alpha^1 \Delta w^1 + ... + \alpha^i \Delta w^i + ... + \alpha^k \Delta w^k, \tag{4.2}$$

where $\alpha^1, ..., \alpha^i, ..., \alpha^k$ are optimizable variables. Considering that we expect to find the model with better robust generalization via optimizing α, a straightforward idea is to optimize α^i with respect to improving its robust performance on a small unseen dataset. That is, we expect WOT carefully selects the optimization trajectories such that the adversarial robustness of the model can better generalize to unseen datasets.

Formally, the **objective function** of optimizing α^i is defined as follows:

$$\min_{0 \le \alpha^i \le 1} \max_{\|\Delta x_{uns}\| \le \epsilon} L(f_{w+\widetilde{\Delta w}}(x_{uns} + \Delta x_{uns}), y_{uns}), \tag{4.3}$$

where (x_{uns}, y_{uns}) is from an unseen dataset and Δx_{uns} is the adversarial perturbations. We constrain α^i to [0,1] such that the new update direction does not go far from the original optimization trajectories and avoid overfitting to the unseen dataset.

Update α^i. α^i can be optimized by any SGD-based optimizers according to the objective function (Eq. (4.3)) described above. In this study, we update α^i

by SGD optimizer with momentum buffer.

$$m^t = m^{t-1} \cdot \gamma + \nabla_{\alpha^i} L(f_{w^{i-1}+\widetilde{\Delta w}}(x_{uns} + \Delta x_{uns}), y_{uns}) \tag{4.4}$$

$$\alpha^i = \alpha^i - lr \cdot m^t, \tag{4.5}$$

where m^t is the momentum buffer of α^i at the t-th step and lr is the learning rate.

4.3.3 WOT: In-Time Refining Optimization Trajectories

Different from adapting optimization trajectories for a post-process like SWA [IPG⁺18], we refine optimization trajectories in time across the training process to encourage the optimization process to go in the direction of better generalization.

Based on whether weight space is considered as a whole or divided into independent blocks, we categorize WOT into two variants: WOT-Whole (WOT-W) and WOT-Blockwise (WOT-B).

WOT-W takes weight space as a whole. It assigns an α for whole weight space and the number of α that need to be optimized equals the number of Gaps:k.

WOT-B considers the weight space in a blockwise way. It assigns an α vector for weight space and the length of the α vector equals the number of blocks. Therefore, Eq. (4.2) can be extended as follows:

$$\widetilde{\Delta w} = \begin{bmatrix} \widetilde{\Delta w}_1 \\ \dots \\ \widetilde{\Delta w}_j \\ \dots \\ \widetilde{\Delta w}_t \end{bmatrix}, \ \widetilde{\Delta w}_j = \alpha_j^1 \Delta w_j^1 + \alpha_j^2 \Delta w_j^2 + \dots + \alpha_j^k \Delta w_j^k \tag{4.6}$$

where j denotes the j-th block. Optimizing α for blockwise of WOT is exactly the same as the description in Eq. (4.4) and Eq. (4.5).

4.4 Experiments

We perform extensive experiments to show the effectiveness of our method in improving adversarial robustness as well as addressing the issue of robust overfitting.

Datasets. Four datasets are considered in our experiments: CIFAR-10, CIFAR-100 [KNH10], Tiny-ImageNet [DDS⁺09] and SVHN [NWC⁺11]. For experiments

Table 4.1: Robust accuracy of WOT under multiple adversarial attacks with various adversarial training variants. The experiments are conducted on CIFAR-10 with the PreRN-18 architecture. The best results are marked in bold.

Models	FGSM	PGD-20	PGD-100	CW_∞	AA-L_∞
AT+early stop	57.30	52.90	51.90	50.90	47.43
AT+SWA	58.89	53.02	51.86	52.32	48.61
AT+WOT-W (Ours)	58.50	53.19	51.90	51.74	48.36
AT+WOT-B (Ours)	**59.67**	**54.85**	**53.77**	**52.56**	**48.96**
Trades	58.16	53.14	52.17	51.24	48.90
Trades+SWA	58.07	53.17	52.22	50.91	49.07
Trades+WOT-W (Ours)	**58.95**	**54.07**	**53.29**	51.74	49.95
Trades+WOT-B (Ours)	58.50	53.73	52.95	**52.12**	**50.19**
MART	59.93	54.07	52.30	50.16	47.01
MART+SWA	58.19	54.21	53.56	49.39	46.86
MART+WOT-W (Ours)	58.13	53.79	52.66	50.24	47.43
MART+WOT-B (Ours)	**59.95**	**55.13**	**54.09**	**50.56**	**47.49**
AT+AWP	59.11	55.45	54.88	52.50	49.65
AT+AWP+SWA	58.23	55.54	54.91	51.88	49.39
AT+AWP+WOT-W (Ours)	59.05	**55.95**	54.96	52.70	49.84
AT+AWP+WOT-B (Ours)	**59.26**	55.69	**55.09**	**52.82**	**50.00**

of WOT, we randomly split 1000 samples from the original CIFAR-10 training set, 10000 samples from Tiny-ImageNet, and 2000 samples from the original CIFAR-100 and SVHN training set as the unseen hold-out sets.

Baselines. Five baselines are included: AT [RWK20], Trades [ZYJ+19a], AWP+AT [WXW20], MART [WZY+20] and SWA [CZL+20]. Three architectures including VGG-16 [SZ14], PreActResNet-18 (PreRN-18) [HZRS16], WideResNet-34-10 (WRN-34-10) [ZK16].

Experimental Setting. For WOT, we adopt an SGD optimizer with a momentum of 0.9, weight decay of 5e-4 and a total epoch of 200 with a batch size of 128 following [RWK20]. By default, we start to refine optimization trajectories after 100 epochs. For WOT-B, we set each block in PreRN-18 and WRN-34-10 architectures as the independent weight space. We set the layers with the same width as a group and set each group as an independent block for VGG-16. We by default set the gaps m to 400, the number of gaps k to 4 and initialize α as zero. For all baselines, we use the training setups and hyperparameters exactly the same as their papers.

Evaluation Setting. We use AA attack [CH20b] as our main adversarial robustness evaluation method. AA attack is a parameter-free ensembled adversarial attack that incorporates three white-box attacks: APGD-CE [CH20b], APGD-T [CH20b], FAB-T [CH20a] and one black-box attack: Square attack [ACFH20]. To the best of our knowledge, AA attack is currently the most reliable adversarial attack for evaluating adversarial robustness. We also adopt three other commonly used white-box adversarial attacks: FGSM [GSS14a], PGD-20/100 [MMS+17] and C&W$_\infty$ attack [CW17b]. Besides, we also report the performance of query-based SPSA black-box attack [UOKO18] (100 iterations with a learning rate of 0.01 and 256 samples for each gradient estimation). By default, we report *the mean of three random runs* for all experiments of our method and omit the standard deviation since it is very small ($\leq 0.3\%$). We by default set $\epsilon = 8/255$ for L_∞ version adversarial attack and $\epsilon = 64/255$ for L_2 version adversarial attack.

Table 4.2: Test robustness under multiple adversarial attacks based on VGG-16/WRN-34-10 architectures. The experiments are conducted on CIFAR-10 with AT and Trades. The bold denotes the best performance.

Architecture	Method	CW$_\infty$	PGD-20	PGD-100	AA-L_∞
VGG16	AT+early stop	46.87	49.95	46.87	43.63
VGG16	AT+SWA	47.01	49.58	49.13	43.89
VGG16	AT+WOT-W(Ours)	47.42	49.96	49.36	44.01
VGG16	AT+WOT-B(Ours)	**47.52**	**50.28**	**49.58**	**44.10**
VGG16	TRADES	45.47	48.24	47.54	43.64
VGG16	TRADES+SWA	45.92	48.64	47.86	44.12
VGG16	TRADES+WOT-W(Ours)	**46.75**	**49.19**	**48.28**	**44.82**
VGG16	TRADES+WOT-B(Ours)	46.21	48.81	47.85	44.17
WRN-34-10	AT+early stop	53.82	55.06	53.96	51.77
WRN-34-10	AT+SWA	**88.45**	55.34	53.61	52.25
WRN-34-10	AT+WOT-W(Ours)	56.05	58.21	57.11	52.88
WRN-34-10	AT+WOT-B(Ours)	**57.13**	**60.15**	**59.38**	**53.89**
WRN-34-10	TRADES	54.20	56.33	56.07	53.08
WRN-34-10	TRADES+SWA	54.55	54.95	53.08	51.43
WRN-34-10	TRADES+WOT-W(Ours)	56.10	57.56	56.20	53.68
WRN-34-10	TRADES+WOT-B(Ours)	**56.62**	**57.92**	**56.80**	**54.33**

Table 4.3: Test robustness under AA-L_2 and AA-L_∞ attacks across various datasets. The experiments are based on PreRN-18 and AT. The bold denotes the best performance.

Attack	Method	SVHN		CIFAR-10		CIFAR-100		Tiny-ImageNet	
		Clean	Robust	Clean	Robust	Clean	Robust	Clean	Robust
L_∞	AT+early stop	89.05	45.72	81.72	47.43	53.84	23.69	42.76	14.39
L_∞	AT+SWA	90.36	40.24	**85.23**	48.61	**58.51**	23.90	49.19	17.94
L_∞	AT+WOT-W(Ours)	**93.25**	50.42	84.47	48.36	55.07	24.41	**49.31**	17.10
L_∞	AT+WOT-B(Ours)	92.95	**51.83**	83.84	**48.96**	54.39	**25.26**	48.83	**18.77**
L_2	AT+early stop	89.05	72.13	81.72	71.30	53.84	42.75	42.76	36.61
L_2	AT+SWA	90.36	67.76	**85.23**	73.28	**58.51**	43.10	49.19	42.40
L_2	AT+WOT-W(Ours)	**93.25**	72.75	84.47	73.20	55.07	**43.88**	**49.31**	42.43
L_2	AT+WOT-B(Ours)	92.95	**72.80**	83.84	**73.39**	54.39	43.32	48.83	**42.54**

4.4.1 Superior Performance in Improving Adversarial Robustness

We evaluate the effectiveness of WOT in improving adversarial robustness across AT and three of its variants, four popular used datasets, i.e., SVHN, CIFAR-10, CIFAR-100 and Tiny-ImageNet, and three architectures, i.e., VGG16, PreRN-18, and WRN-34-10.

WOT consistently improves the adversarial robustness of all adversarial training variants. In Table 4.1, we applied WOT-B and WOT-W to AT+early stop, Trades, MART, and AWP variants and compare them with their counterpart baselines. Besides, we add the combination of SWA and these adversarial training variants as one of the baselines. The results show: **(1)** WOT consistently improves adversarial robustness among the four adversarial training variants under both weak attacks, e.g. FGSM, PGD-20, and strong attacks, e.g., C&W$_\infty$, AA-L_∞ attacks. **(2)** WOT-B as the WOT variant confirms our hypothesis and consistently performs better than WOT-W. WOT-B improves the robust accuracy over their counterpart baselines by 0.35% ~ 1.53% under AA-L_∞ attack. **(3)** WOT boosts robust accuracy with a larger margin on AT and Trades than MART and AWP under AA-L_∞ attack. One reason might be that MART and AWP themselves enjoy good ability in mitigating robust overfitting [SHS21, WXW20], leading to less space for WOT to further boost the performance.

WOT can generalize to different architectures and datasets. Table 4.2 and Table 4.3 show that WOT consistently outperforms the counterpart baseline under AA-L_∞ attack, which indicates that the effectiveness of WOT generalizes well to different architectures and datasets. In Table 4.2, WOT boosts robust accuracy by 0.47% ~ 2.12% on VGG16 and WRN-34-10 architectures. In Table 4.3, WOT improves robust accuracy with 1.53% ~ 6.11% among SVHN, CIFAR-10, CIFAR-100 and Tiny-ImageNet under AA-L_∞ attack. Besides, the success of WOT can also be extended to AA-L_2 attack with the improvement by 0.67% ~ 5.93%.

Excluding Obfuscated Gradients. Athalye et al. [ACW18] claims that obfuscated gradients can also lead to the “counterfeit" of improved robust accuracy under gradients-based white-box attacks. To exclude this possibility, we report the performance of different checkpoints under transfer attack and SPSA black-box attack over epochs. In Figure 4.3, the left figure shows robust accuracy of the unseen robust model on the adversarial examples generated by the PreRN-18 model trained by AT, AT+WOT-B, AT+WOT-W respectively with PGD-10 attack on CIFAR-10. A higher robust accuracy on the unseen robust model corresponds to a weaker attack. It can be seen that both AT+WOT-B and AT+WOT-W generate more transferable adversarial examples than AT. Similarly, the middle figure shows the robust accuracy of the PreRN-18 model trained by AT, AT+WOT-B, and AT+WOT-W on the adversarial examples generated by the unseen robust model. It can be seen that AT+WOT-B, and AT+WOT-W can better defend the adversarial examples from the unseen model. What’s more, in the right figure, we observe again that both AT+WOT-B and AT+WOT-W outperform AT under SPSA black-box attack over different checkpoints during training. All these empirical results sufficiently suggest that the improved robust accuracy of WOT is not caused by obfuscated gradients.

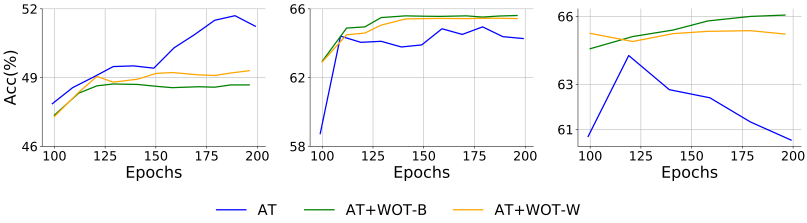


Figure 4.3: Robust accuracy under black-box attack over epochs. **(Left)** Robust accuracy on the unseen robust model transfer attacked from checkpoints of AT, AT+WOT-W/B. **(Middle)** Robust accuracy on checkpoints of AT, AT+WOT-W/B transfer attacked from the unseen model. **(Right)** Robust accuracy on checkpoints of AT, AT+WOT-W/B under SPSA black-box attack. The experiments are conducted on PreRN-18 and CIFAR-10. The unseen robust model is WRN-34-10 trained by AT.

4.4.2 Ability to Prevent Robust Overfitting

We report the robust accuracy under AA-L_∞ attack for the best checkpoint and the last checkpoint based on PreRN-18 and WRN-34-10 architectures on CIFAR-10 (Table 4.4). Besides, we show the robust accuracy curve under PGD-10 attack on different checkpoints over epochs (Figure 4.4).

In Figure 4.4, the third and fourth figures show that after the first learning rate decay (at 100 epoch), there is a large robust accuracy drop for AT between the best checkpoint and the last checkpoint on both PreRN-18 and WRN-34-10 architectures. In comparison, there is completely no robust accuracy drop for AT+WOT-W/B between the best checkpoint and the last checkpoint on both PreRN-18 and WRN-34-10 architectures. In Table 4.4, we further show the evidence that there is no robust accuracy drop for AT+WOT-B/W under stronger attack, i.e., AA-L_∞ attack. From the first and second figures of Figure 4.4, we observe that the mean of α decreases to a very small value after 150,100 epochs for PreRN-18, WRN-34-10 respectively. The small mean of α indicates that WOT stops the model's weights updating with unexpected magnitudes, which prevents the occurrence of robust overfitting.

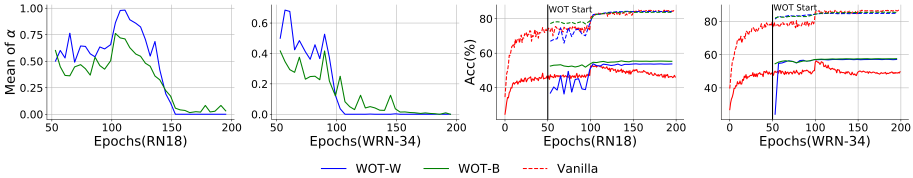


Figure 4.4: Mean value of α and results of test robust/clean accuracy over epochs. The experiments are conducted on CIFAR-10 with PreRN-18 based on AT.

4.4.3 Ablations and Visualizations

In this section, we first conduct ablation studies to show the effectiveness of the designed optimization trajectories and the unseen hold-out set in WOT. Then we investigate the impact of gaps:m and the number of gaps:k, the effect of WOT on the loss landscapes w.r.t weight space, and the visualization of α for blocks. The results are shown in Table 4.5, Figure 4.5, and Figure 4.7. All experiments in the two figures are conducted on CIFAR-10 with PreRN-18 based on AT except for Figure 4.7 where Trades is also included. The robust accuracy is evaluated under AA-L_∞ attack for all three figures.

Table 4.4: Test robustness under AA-L_∞ attack to show the issue of robust overfitting in AT and the effectiveness of WOT in overcoming it. The difference between the best and final checkpoints indicates performance degradation during training and the best checkpoint is chosen by PGD-10 attack on the validation set. The experiments are conducted on CIFAR-10 with PreRN-18/WRN-34-10 architectures.

Architectures	Method	Robust Accuracy(RA)			Standard Accuracy(SA)		
		Best	Final	Diff.	Best	Final	Diff.
PreRN-18	AT	48.02	42.48	-5.54	81.33	84.40	+3.07
PreRN-18	AT+SWA	48.93	48.61	-0.32	84.19	85.23	+1.04
PreRN-18	AT+WOT-W(Ours)	48.04	48.36	+0.32	84.05	84.47	-0.42
PreRN-18	AT+WOT-B(Ours)	48.90	48.96	+0.06	83.84	83.83	-0.01
WRN-34-10	AT	51.77	46.78	-4.99	85.74	86.34	+0.6
WRN-34-10	AT+SWA	53.38	52.25	-1.13	87.14	88.45	+1.31
WRN-34-10	AT+WOT-W(Ours)	52.84	52.88	+0.04	84.83	84.88	+0.05
WRN-34-10	AT+WOT-B(Ours)	52.23	53.89	+1.66	83.46	85.50	+2.04

Table 4.5: Robust Accuracy of ablation experiments on CIFAR-10 with PreRN-18.

Methods	PGD-20	PGD-100	CW$_\infty$	AA-L$_\infty$
AT+B1	49.68	47.44	49.04	45.26
AT+B2	52.74	51.28	51.31	48.22
AT+WOT-B+B3 (m=400,k=4)	47.14	44.23	43.87	41.02
AT+WOT-W (m=400,k=4)	53.19	51.90	51.74	48.36
AT+WOT-B (m=400,k=4)	54.85	53.77	52.56	48.96

Ablation studies. To demonstrate the effectiveness of the designed optimization trajectories and the unseen hold-out set in boosting adversarial robustness, we designed the following baselines: 1) Keep the same unseen hold-out set and training strategy with WOT, but optimize model weights instead of α on the unseen hold-out set (Abbreviated as “B1"); 2) Keep the same unseen hold-out set and optimize the hyperparameter of SWA by the hold-out set (Abbreviated as “B2"); 3) Replace the unseen hold-out set with a seen set, i.e. keep the same number of samples from the training set (Abbreviated as “B3"). Results in Table 4.5 show that **(1)** AT+WOT-W/B outperforms AT+B1 and AT+B2, indicating the designed optimization trajectories play key roles in WOT. **(2)** AT+WOT-W/B outperforms AT+B3 with a large margin, indicating the unseen hold-out set is crucial for WOT.

Impact of *m* and *k*. Figure 4.5 shows the impact of gaps *m* and number of

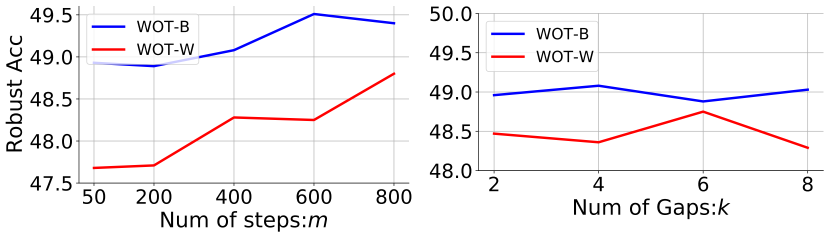
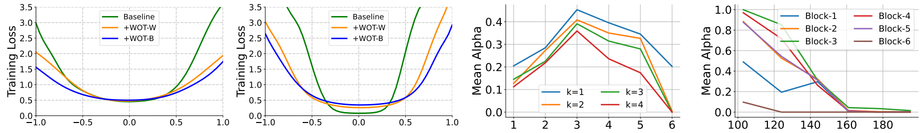


Figure 4.5: The impact of gaps m and the number of gaps k on robust accuracy under AA-L_∞ attack. The experiments are conducted on CIFAR-10 with PreRN-18 based on AT. k is fixed to 4 for the left figure and m is fixed to 400 for the right figure.



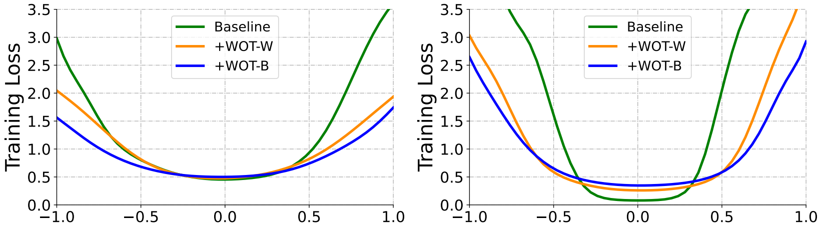
(a) Trades (b) AT (c) Blocks (X-axis) (d) Epochs (X-axis)

Figure 4.6: Loss landscape w.r.t weight space (Figure 4.6a and Figure 4.6b). z-axis denotes the loss value. We plot the loss landscape following the setting in [WXW20]. The averaged α by averaging along training process (Figure 4.6c). The k-averaged α during the training process. (Figure 4.6d). The experiments are conducted on CIFAR-10 with PreRN-18.

gaps k on robust accuracy under AA-L_∞ attack. In the left figure, we observe that robust accuracy increases with an increase of m. Besides, we find that WOT-W is more sensitive to m than WOT-B. The right figure shows that both WOT-W and WOT-B are not sensitive to the number of gaps k.

Averaged α for Blocks. To shed insights on why WOT-B outperforms WOT-W, we plot the learned α for each block. Experiments are conducted on CIFAR-10 with PreRN-18 based on WOT-B (K=4, m=400). Results in Figure 4.6c and Figure 4.6d show that the magnitude of learned α is different among blocks. Specifically, WOT-B assigns a large value of α for middle blocks, i.e., Block-2,3,4,5, and a small value of α for the bottom and top blocks, i.e., Block-1,6. This indicates that assigning different weights for different blocks may play a crucial role in boosting adversarial robustness.

Visualizing loss landscape. We expect WOT to search flatter minima for adversarial training to boost its robust generalization. We demonstrate that it



(a) +Trades

(b) +AT

Figure 4.7: Loss landscape w.r.t weight space. z-axis denotes the loss value. We plot the loss landscape following the setting in [WXW20] with $z = loss(f_{w+x\cdot v}(i))$ where v is sampled from Gaussian distribution and i denotes the inputs. The experiments are conducted on CIFAR-10 with PreRN-18.

indeed happens via visualizing the loss landscape with respect to input space (Figure 4.8) and weight space (Figure 4.7). Figure 4.8 shows that WOT enjoys a loss landscape with low curvature compared with AT, which is in line with the robust generalization claim in [MDFUF18]. Figure 4.7 shows that WOT obtains flatter minima than AT, which indicates an improved robust generalization [SHS21,WXW20].

4.5 Conclusion

In this chapter, we proposed a new method named weighted optimization trajectories (WOT) for improving adversarial robustness and avoiding robust overfitting. We re-weighted the optimization trajectories in time by maximizing the robust performance on an unseen hold-out set during the training process. The comprehensive experiments demonstrated: (1) WOT can effectively improve adversarial robustness across various adversarial training variants, model architectures, and benchmark datasets. (2) WOT exhibits superior performance in mitigating robust overfitting. Moreover, visualizing analysis validates that WOT flattens the loss landscape with respect to input and weight space, showing an improved robust generalization.

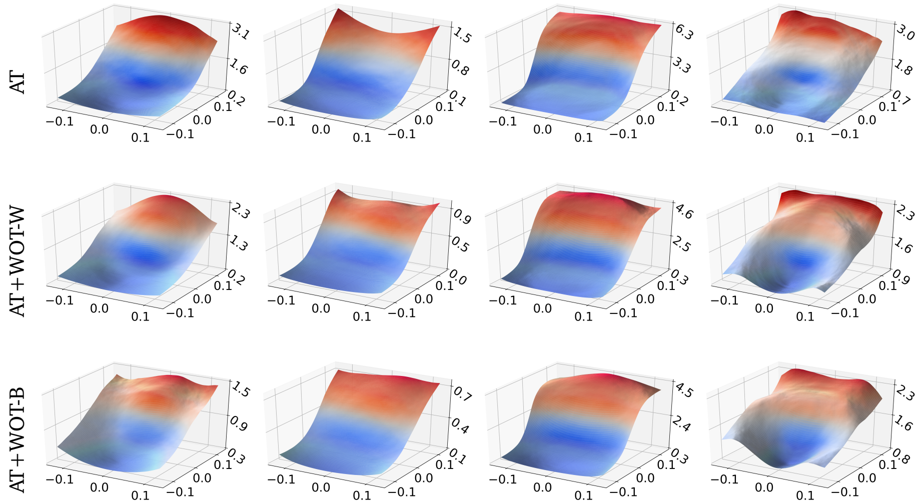


Figure 4.8: Comparison of loss landscapes of PreRN-18 models trained by AT (the first row) and our methods (the second and third row). Loss plots in each column are generated from the same original image randomly chosen from the CIFAR-10 test dataset. z-axis denotes the loss value. Following the setting in [EIA18], we plot the loss landscape function: $z = loss(x \cdot r_1 + y \cdot r_2)$ where $r_1 = sign(\nabla_x f(x))$ and $r_2 \sim Rademacher(0.5)$.

Chapter 5

Bridging the Performance Gap between FGSM and PGD Adversarial Training

Deep learning has demonstrated impressive performance in many tasks, but is susceptible to adversarial examples. Adversarial training with the projected gradient decent (*adv.PGD*) is considered one of the most effective defense techniques, but it requires a significant amount of training time due to the need for multiple iterations to generate perturbations. On the other hand, adversarial training with the fast gradient sign method (*adv.FGSM*) is faster, as it only requires one step to generate perturbations, but does not provide enough adversarial robustness. In this chapter, we extend *adv.FGSM* to make it achieve a comparable adversarial robustness with *adv.PGD*. We uncover the reason for the difference in adversarial robustness between *adv.FGSM* and *adv.PGD*, which lies in the large curvature along the FGSM perturbed direction. To address this issue, we propose combining *adv.FGSM* with a curvature regularization (*adv.FGSMR*) in order to bridge the performance gap between *adv.FGSM* and *adv.PGD*. The experiments show that *adv.FGSMR* has higher training efficiency than *adv.PGD*. In addition, it achieves a comparable performance of adversarial robustness on the MNIST dataset under white-box attack, and it achieves better performance than *adv.PGD* under white-box attack and effectively defends the transferable adversarial attack on the CIFAR-10 dataset.

5.1 Introduction

Deep Neural Networks (*DNNs*) have shown great performance in multiple tasks, e.g. image classification [KH12, HZRS16], object detection [GDDM14], semantic segmentation [LSD15], and speech recognition [HDY⁺12]. However, these highly performed models show weakness in adversarial examples. Namely, carefully designed imperceptible perturbations on input can change the prediction drastically [SZS$^+$13, GSS14a]. This fragility prohibits *DNNs* to be widely applied especially in security-sensitive tasks such as autonomous cars, face recognition, and malware detection. Therefore, training a model resistant to adversarial attacks becomes increasingly important.

By now, plenty of ways have been proposed to generate adversarial examples, which can be categorized into black-box attacks and white-box attacks. White-box attacks can access the complete knowledge of the target model including its parameters, architecture, training method, and training data. The popular white-box attacks include *FGSM* [GSS14a], *PGD* [MMS$^+$18], *Deepfool* [MDFF16], *C&W* [CW17b], etc. Black-box attack generates adversarial examples without knowledge of the target model, e.g. *ZOO* [CZS$^+$17], *Transferable adversarial attack* [LCLS17, PMG$^+$17], etc. Correspondingly, many methods have been proposed to improve the model's adversarial robustness against these attacks. Qiu et al. [QLZW19], Akhtar and Mian [AM18] separate these defense methods into three categories: (1) augmenting training data, e.g. adversarial training [MMS$^+$18, GSS14a]; (2) using an extra tool to help model against adversarial attacks, e.g. PixelDefend [SKN$^+$17]; and (3) modifying model to improve its robustness, e.g. Defensive Distillation [PMW$^+$16], Regularization [MDFUF18, JG18].

Among these defense approaches, most have been reported failure on later proposed adversarial attacks except for adversarial training [ACW18]. *adv.PGD* has been considered one of the most effective ways to achieve moderate adversarial robustness [WMB$^+$19]. However, a major issue for *adv.PGD* is its expensive computational cost because *PGD* attack takes multi-step iterations to generate perturbations. The high computational cost makes this method hard to be applied to larger neural networks and datasets. On the other hand, *adv.FGSM* takes much less computational cost but shows no robustness improvement against adversarial attacks except for *FGSM* attack (Table 5.1). The behavior of strong defense on *FGSM* attack but a weak defense on other attacks has also been reported in [MMS$^+$18, KGB17]. We believe that closing the robust performance gap between *adv.FGSM* and *adv.PGD* would bring significant value. Our investigation into the lack of adversarial robust performance in *adv.FGSM* has revealed that the large curvature along the *FGSM* perturbed direction results in a significant disparity between the perturbed directions generated by *FGSM* and *PGD* attacks, which accounts for the differences in robustness between *adv.FGSM* and *adv.PGD*(Figure 5.1). To deal with this we propose a regularization term that makes *FGSM* perturbed direction close to *PGD* perturbed direction, and allows for *adv.FGSM* to reach comparable robust performance as *adv.PGD*. Our experimental studies demonstrate that the proposed method achieves comparable results on the MNIST dataset and better results on the CIFAR-10 dataset compared with *adv.PGD*.

Our contributions are summarized as follows:

- We analyze the influence of the curvature along *FGSM* perturbed direction on the perturbations generated by *FGSM* and *PGD* attacks respectively. We show that the curvature along *FGSM* perturbed direction has a significant influence on the performance of adversarial robustness achieved by *adv.FGSM*.

- We develop a curvature regularization term for restraining the curvature along *FGSM* perturbed direction when training model with *adv.FGSM*, which is called *adv.FGSMR* method. *adv.FGSMR* can effectively bridge the performance gap between *adv.FGSM* and *adv.PGD*.

- Extensive experiments show that *adv.FGSMR* achieves comparable performance on MNIST under a white-box attack. Besides, it achieves better performance on CIFAR-10 under white-box attack and effectively defends the transferable adversarial attack as well. Experiments also show that *adv.FGSMR* achieves comparable convergence speed on perturbed-data accuracy during the training process while requiring only half of the time for training one epoch compared with *adv.PGD*.

The rest of this paper is organized as follows. Section 5.2 describes the preliminary knowledge. Section 5.3 presents the proposed method for bridging the performance gap between *adv.FGSM* and *adv.PGD*. Section 5.4 introduces evaluations in terms of training efficiency and adversarial robustness.Finally, Section 5.5 draws the conclusions of this study.

5.2 Preliminaries

5.2.1 Notation

Method	*PGD-l2*	*PGD-inf*	*Deepfool-l2(ρ_{adv})*	*C&W(ρ_{adv})*
adv.PGD	0.710	0.444	0.178	0.129
adv.FGSM	0.353	0.091	0.022	0.016

Table 5.1: Comparison of robust performance of robust models trained by *adv.FGSM* and *adv.PGD* respectively against various attacks. Experiments are based on the CIFAR-10 test set and *ResNet-18* model. For *Deepfool-l2* and *C&W-l2* attacks, ρ_{adv} is calculated using Eq. (5.5).

We denote our deep neural network as $f_\theta(x)$ where $x \in R^d$ is an instance of input data, and $L(f_\theta(x), y)$ is the *cross-entropy* loss where y is the true label. $sgn(\cdot)$ denotes the sign function. $\nabla_x L(\cdot)$ denotes the gradient of $L(\cdot)$ with respect to x. S is the set constrained by l_∞ or l_2 ball. ϵ is the allowed perturbation size. k is the total iterations for *PGD* attacks.

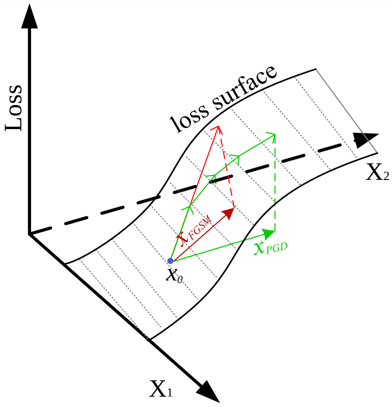


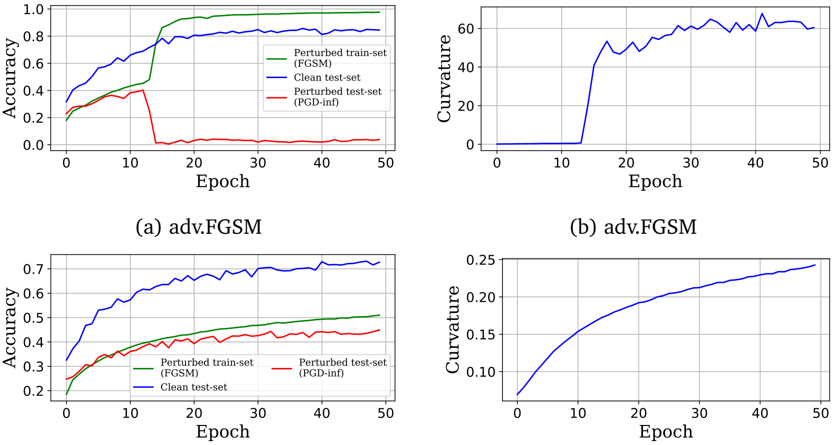
Figure 5.2: The simplified schematic diagram for perturbed directions generated by *PGD-inf* and *FGSM* attacks. x_0 is a specific input.

5.2.2 Adversarial Training

Different from *Vanilla training*, adversarial training uses adversarial samples instead of clean samples to train a model. Generally, the optimization function of adversarial training can be represented as follows [MMS⁺18]:

$$\min_\theta \rho(\theta), \rho(\theta) = \mathbb{E}_{(x,y)\sim D}[\max_{\delta \in S} L(f_\theta(x+\delta), y)]. \tag{5.1}$$

In this chapter, we call it *adv.PGD* method when $\max_{\delta \in S} L(f_\theta(x+\delta), y)$ is solved by *PGD-inf* attack. Similarly, we call it *adv.FGSM* method when $\max_{\delta \in S} L(f_\theta(x+\delta), y)$ is solved by *FGSM* attack. It is easy to see that *adv.PGD* takes much more training time than *adv.FGSM* since *PGD-inf* attack takes multiple iterations while *FGSM* attack takes only one iteration.



(a) adv.FGSM

(b) adv.FGSM

(c) adv.FGSMR

(d) adv.FGSMR

Figure 5.1: The accuracy and average curvature curve for training *ResNet-18* model on CIFAR-10 within 50 epochs. The subfigure *(a)* and *(b)* show the accuracy and average curvature curve of the model trained by *adv.FGSM* respectively; *(c)* and *(d)* show the accuracy and average curvature curve of the model trained by our proposed *adv.FGSMR* respectively. The curvature value is calculated using Eq. (5.3). Notice: A sudden decrease of perturbed accuracy under *PGD-inf* attack occurs with the sudden increase of the curvature value for *adv.FGSM*.

5.3 Methodology

In this section, we first give a full analysis to explain why *adv.FGSM* can not achieve the performance of adversarial robustness with *adv.PGD*. Based on the analysis, we further extend *adv.FGSM* in order to achieve comparable performance with *adv.PGD*.

5.3.1 Analysis of Performance Gap between adv.FGSM and adv.PGD

Considering the only difference between *adv.FGSM* and *adv.PGD* is that the adversarial examples are generated by *FGSM* attack or *PGD-inf* attack. Thus we first mainly explore the perturbation difference generated by *FGSM* and *PGD-inf* respectively. From the definitions of *PGD-inf* and *FGSM* attacks in Section 5.2, we know *PGD-inf* attack is a multi-step variant of *FGSM* attack and it is apparent that *PGD-inf* attack can generate more accurate perturbation compared with *FGSM* attack. Figure 5.2 shows the simplified schematic of *PGD-inf* and *FGSM* attacks. It indicates that the difference of perturbed directions generated by them will be enlarged with the increasing of the curvature along *FGSM* perturbed direction. We believe that a large difference in perturbed directions will lead to the radical difference in adversarial robust performance achieved by *adv.FGSM* and *adv.PGD* because the perturbed training dataset depends on these perturbed directions. This conjecture is supported by the experiment in Figure 5.3a. Therefore, we propose that as long as the curvature along *FGSM* perturbed direction is kept to be small during the training process, *adv.FGSM* can achieve comparable performance with *adv.PGD* for the following reasons:

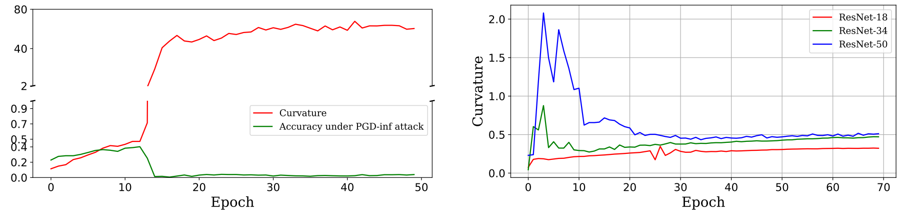
- The perturbed directions generated by *FGSM* and *PGD-inf* attacks will be approaching to be identical with the curvature along *FGSM* perturbed direction approaching zero (Figure 5.2). As soon as the perturbed directions are the same, the perturbed training set will also be the same since the size of perturbation has the same constraint, and consequently the performance of adversarial robustness between *adv.FGSM* and *adv.PGD* should be the same.

- During *adv.FGSM* training process, the perturbed-data accuracy under *PGD-inf* attack stops increasing until the curvature along *FGSM* perturbed direction surges suddenly (Figure 5.3a). This provides evidence that the curvature along *FGSM* has a significant influence on the adversarial robust performance of *adv.FGSM*.

- The curvature along *FGSM* perturbed direction is also kept to be small during *adv.PGD* training process (Figure 5.3b). It indicates that restraining the growth of the curvature value is reasonable.

5.3.2 Proposed Method

We use a curvature regularization for restraining the growth of the curvature value and making *FGSM* perturbed direction close to *PGD-inf* perturbed direction. Formally, Let $L_\theta(x)$ be the cross-entropy loss; $g = sgn(\nabla_x L_\theta(x))$ be the *FGSM* perturbed direction at data point x; $\delta = \epsilon g$ be the perturbation generated by *FGSM* attack. As what we want to restrain is the gradient variation along *FGSM*



(a) adv.FGSM

(b) adv.PGD

Figure 5.3: *(a)* The average curvature along *FGSM* perturbed direction on the CIFAR-10 training set and the perturbed-data accuracy curve on perturbed test set generated by *PGD-inf* attack. The training process is based on *ResNet-18* model and *adv.FGSM*. *(b)* The average curvature along *FGSM* perturbed direction on the CIFAR-10 training set during the training process with *adv.PGD*. The curvature value is calculated using Eq. (5.3).

perturbed direction, namely, the second directional derivative, here we want to emphasize that the curvature value corresponds to the second directional derivative instead of the exact definition of curvature. According to the definition of the directional derivative, the second derivative along *FGSM* perturbed direction can be represented as:

$$\nabla^2_{xg} L_\theta(x) = \lim_{\epsilon \to 0} \frac{\nabla_x L_\theta(x + \epsilon g) - \nabla_x L_\theta(x)}{\epsilon}. \tag{5.2}$$

Following the paper [MDFUF18], by using a finite difference approximation, we have $\nabla^2_{xg} L_\theta(x) = \frac{\nabla_x L_\theta(x+\epsilon g) - \nabla_x L_\theta(x)}{\epsilon}$. The denominator can be omitted since it is a constant. Therefore, we give the curvature regularization term as follows:

$$R_\theta = \|\nabla_x L_\theta(x + \epsilon g) - \nabla_x L_\theta(x)\|_2, \tag{5.3}$$

The form of Eq. (5.3) is similar to *CURE* method [MDFUF18] but a difference is that the perturbation size here is fixed and the perturbed direction is generated by *FGSM* attack. The adversarial training optimization goal is to minimize the following expression:

$$\min_\theta L_\theta(x + \epsilon g) + \lambda R_\theta, \tag{5.4}$$

where R_θ is the curvature regularization defined in Eq. (5.3). λ is the hyperparameter to control the strength of penalizing the curvature along *FGSM* perturbed direction.

5.4 Experiments

5.4.1 Experiments Setup

Datasets and network architectures All experiments are run on the MNIST and CIFAR-10 datasets. MNIST [LBB+98] consists of 28x28 gray-scale images for handwritten digits with 60K training images and 10K test images. CIFAR-10 [KH+09] consists of 32x32 color images that contain 10 different classes with 50K training images and 10K test images.

For the MNIST dataset, we use a simple convolutional neural network with four convolutions and two dense layers as our model architecture. For the CIFAR-10 dataset, the Residual Networks-18/34/50 [HZRS16] and Wide Residual Networks-22 × 1/5/10 × 0 × 10 [ZK16] are used as our model architecture. For comparison, robust models trained by *adv.PGD* and *adv.FGSM* respectively are evaluated as well. Please refer to the supplementary material for a detailed training process.

Adversarial attacks In order to have a comprehensive evaluation of the model's robustness, state-of-the-art white-box attacks are employed here. In specific, *PGD-inf*, *PGD-l2*, *FGSM*, *C&W-l2* and *Deepfool-l2* are used for white-box attack. By default, the hyperparameter k is set to 20 for *PGD-inf/l2* in this chapter. The accuracy on the perturbed test set is used as the adversarial robustness indicator, but for *C&W-l2* and *Deepfool-l2* attacks, as they can find the adversarial examples that change the model's prediction for all inputs, we use the distance of the perturbed example to the clean example as the adversarial robustness evaluation indicator, refer to [MDFF16], the average distances are defined as follows:

$$\rho_{adv} = \frac{1}{|\mathscr{D}|} \sum_{x \in \mathscr{D}} \frac{\|x_{adv} - x\|_2}{\|x\|_2}, \tag{5.5}$$

where x_{adv} is the adversarial example generated by the attack algorithm, and $\mathscr{D}$ is the test set. *C&W-l2* and *Deepfool-l2* attack are carried out by public attack tool: *foolbox* [RBB17] and parameters are set by default for these two attacks. Beyond white-box attacks, the transferable adversarial attack [LCLS17] is employed on the CIFAR-10 dataset as a block-box attack evaluation.

5.4.2 Training Efficiency

We evaluate the training efficiency of *adv.FGSMR* and compare it with *adv.PGD*. The training efficiency is evaluated from two aspects: (1) how much time it takes for training one epoch; and (2) how fast can the adversarial robustness be improved during the training process. For the first aspect, as *adv.PGD* method uses *PGD-inf* attack to generate perturbed examples, it takes k (commonly k is set to 20) times of forward and backward process where k is the total iterations for *PGD-inf* attack. But for *adv.FGSMR*, it takes 1 time of forward and backward process to generate perturbed examples plus 2 times of forward and backward process for the curvature regularization. Therefore, from the analysis above, *adv.FGSMR* saves $(k-3)$ times the forward and backward process. Table 5.2 shows the training time of 50 epochs for *adv.PGD* $(k = 20)$ and *adv.FGSMR* respectively, which indicates that *adv.FGSMR* takes half time of what *adv.PGD* $(k = 20)$ takes. For the second aspect, we record the perturbed-data accuracy under *PGD-inf* attack on the CIFAR-10 test set with the first 50 training epochs for *adv.PGD* and *adv.FGSMR* with $\epsilon = 8.0/255$ respectively. We repeat the training process 10 times and report the mean and standard deviation. The results (Figure 5.4) show that the perturbed-data accuracy of *adv.FGSMR* can be converged as fast as *adv.PGD*. Therefore, we conclude that *adv.FGSMR* has higher training efficiency since it takes less time for training one epoch and has comparable convergence speed upon the perturbed-data accuracy compared with *adv.PGD*.

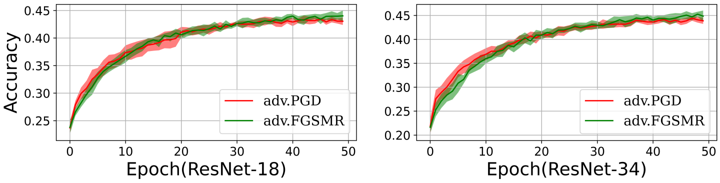


Figure 5.4: A comparable convergence speed on perturbed-data accuracy between *adv.FGSMR* and *adv.PGD*. Left figure: the training process of *ResNet-18* model. Right figure: the training process of *ResNet-34* model. Perturbed test sets are generated by *PGD-inf* attack $(\epsilon = 8.0/255)$ on the CIFAR-10 test set. The accuracy variation for each epoch is plotted using one standard deviation.

5.4.3 Performance under White-box Attack

Performance on the MNIST Dataset We evaluate the performance of our proposed *adv.FGSMR* on the MNIST dataset. For comparison, the performance of *adv.PGD*, *adv.FGSM* and *CURE* [MDFUF18] are shown. Robust models with $\epsilon = 0.1$ and $\epsilon = 0.2$ are trained by *adv.PGD*, *adv.FGSM* and *adv.FGSMR* respectively. Various state-of-the-art attacks are used for evaluating adversarial robustness

Time (minutes)	ResNet-18 (adv.PGD)	ResNet-34 (adv.PGD)	ResNet-18 (Ours)	ResNet-34 (Ours)
Training time(50 Epoch)	214	375	106	187

Table 5.2: Comparison of time spent on training 50 epochs with *adv.PGD* and *adv.FGSMR* respectively. This experiment is based on the CIFAR-10 dataset.

including *FGSM*, *PGD-l2*, *PGD-inf*, *Deepfool-l2* and *C&W-l2* attacks. The hyperparameter ϵ is set to 0.2, 2, 0.1 for *FGSM*, *PGD-l2* and *PGD-inf* attacks respectively.

From Table 5.3, We can see that our method achieves higher perturbed-data accuracy than *adv.PGD* under *FGSM*, *PGD-l2* and *PGD-inf* attacks. For *Deepfool-l2* attack, the average distance ρ_{adv} values of our method are slightly smaller than that of *adv.PGD*. For *C&W-l2* attack, our method achieves a slightly larger distance on the robust model with $\epsilon = 0.2$ while achieving a slightly smaller distance on the robust model with $\epsilon = 0.1$. It is also worth noting that our method achieves state-of-the-art accuracy on the clean test set. In general, our method achieves comparable adversarial robust performance compared with *adv.PGD*.

For *adv.FGSM*, it achieves better performance on *FGSM* attack but performs worse on the other four attacks, which is consistent with the results reported in [KGB17]. Considering the curvature regularization is similar to *CURE* method [MDFUF18], we also show the performance of *CURE* method that is proposed to improve robustness by decreasing the curvature of the loss function. The results (Table 5.3) show that the performance achieved by *CURE* is obviously worse than the performance achieved by *adv.PGD* and *adv.FGSMR* under all attacks.

Performance on the CIFAR-10 Dataset We show the adversarial robust performance of the proposed *adv.FGSMR* on the CIFAR-10 dataset. For comparison, the adversarial robust performance of *adv.PGD* and *Vanilla train* are also evaluated. For *adv.FGSMR*, we train three robustness models with ϵ = 8.0/255, 9.0/255, 10.0/255 respectively. The same as on the MNIST dataset, *FGSM*, *PGD-l2*, *PGD-inf*, *Deepfool-l2* and *C&W-l2* attacks are chosen for testing the adversarial robust performance. The hyperparameter ϵ is set to 8.0/255 for *FGSM* and *PGD-inf* attacks, and 60.0/255 for *PGD-l2* attack.

The results (Table 5.4) show that our method achieves higher perturbed-data accuracy than *adv.train-PGD* under *FGSM*, *PGD-inf* and *PGD-l2* attacks, and the average distance ρ_{adv} values are larger than that of *adv.PGD* under *Deepfool-l2* and *C&W-l2* attacks. The large average distance ρ_{adv} values indicate that our

Attack methods	Clean	*FGSM*	*PGD-l2*	*PGD-inf*	*Deepfool-l2*	*C&W-l2*
Training methods	(accuracy)	(accuracy)	(accuracy)	(accuracy)	(ρ_{adv})	(ρ_{adv})
Vanilla train	0.98	0.361	0.448	0.27	0.54	0.46
adv.PGD(ϵ : 0.1)	0.993	0.897	0.956	0.974	1.25	0.85
adv.PGD(ϵ : 0.2)	0.992	0.966	0.975	0.982	1.36	0.87
adv.FGSM(ϵ : 0.1)	0.992	0.988	0.950	0.971	1.02	0.77
adv.FGSM(ϵ : 0.2)	0.993	0.968	0.950	0.972	1.07	0.66
CURE [MDFUF18]	0.990	0.936	0.932	0.957	1.02	0.79
adv.FGSMR(ϵ : 0.1)	0.994	0.961	0.959	0.979	1.15	0.84
adv.FGSMR(ϵ : 0.2)	0.992	0.968	0.976	0.983	1.31	0.90

Table 5.3: Performance of models trained by *Vanilla train*, *adv.PGD*, *CURE*, *adv.FGSMR* methods respectively against various attacks on the MNIST Dataset. For *FGSM*, *PGD-l2*, and *PGD-inf* attacks, the accuracy on the perturbed MNIST test set is taken as the evaluation indicator. For *Deepfool-l2* and *C&W-l2* attacks, the average distance (ρ_{adv}) is taken as the evaluation indicator and is calculated using Eq. (5.5).

method indeed enlarges the distance of input x to its nearest boundary. For *adv.FGSM*, it achieves much higher accuracy on *FGSM* perturbed examples than on clean examples, which is claimed as the 'label leaking' problem in [KGB17]. The average distance ρ_{adv} also shows that the model trained by *adv.FGSM* nearly does not enlarge the distance of input x to the nearest decision boundary.

We also observe that with increasing perturbation size ϵ from 8.0/255 to 10.0/255, the clean accuracy decreases gradually and the perturbed-data accuracy under *PGD-inf* attack increases gradually, which is consistent with the claim [TSE$^+$18a] that there is a trade-off between clean accuracy and adversarial robustness. However, it is interesting that the perturbed-data accuracy under *FGSM* and *PGD-l2* attacks does not show an increasing trend. We argue the perturbed-data accuracy might depend more on clean accuracy since the *FGSM* and *PGD-l2* attacks are weaker than the *PGD-inf* attack.

Effect of network capacity In order to explore the relation between network capacity and adversarial robustness improved by *adv.FGSMR* ($\epsilon = 8.0/255$), we evaluate the adversarial robust performance on *ResNet-18/34/50* and *Wide ResNet-22×1/5/10×0*×10 for different depths and widths respectively. Madry [MMS$^+$18] concludes by experiments that increasing the capacity of a model can increase the model's adversarial robustness. In our results (Table 5.5), the perturbed-data accuracy achieved by *adv.FGSMR* shows the same increasing tendency both with the increasing of the model's width or depth, which is consistent with the claim of [MMS$^+$18]. Besides, the perturbed-data accuracy

Attack methods	Clean	*FGSM*	*PGD-l2*	*PGD-inf*	*Deepfool-l2*	*C&W-l2*
Training methods	(accuracy)	(accuracy)	(accuracy)	(accuracy)	(ρ_{adv})	(ρ_{adv})
Vanilla train	0.909	0.237	0.308	0.000	0.031	0.025
adv.FGSM$(\epsilon: 8.0/255)$	0.849	0.908	0.353	0.091	0.022	0.016
adv.PGD$(\epsilon: 8.0/255)$	0.746	0.506	0.710	0.444	0.178	0.129
adv.FGSMR$(\epsilon: 8.0/255)$	0.789	0.51	0.759	0.458	0.228	0.179
adv.FGSMR$(\epsilon: 9.0/255)$	0.772	0.507	0.734	0.465	0.227	0.180
adv.FGSMR$(\epsilon: 10.0/255)$	0.756	0.509	0.723	0.470	0.230	0.177

Table 5.4: Performance of models trained by *Vanilla train*, *adv.train-PGD*, *adv.train-FGSMR* methods respectively under various attacks on the CIFAR-10 dataset. For *FGSM* and *PGD-inf/l2* attacks, the accuracy on the perturbed CIFAR-10 test set is taken as the evaluation indicator. For *Deepfool-l2* and *C&W-l2* attacks, the average distance (ρ_{adv}) is taken as evaluation indicator.

achieved by our method is all higher than the perturbed-data accuracy achieved by *adv.PGD*, which further provides evidence that the proposed method achieves better performance on the CIFAR-10 dataset. We also calculate the average curvature for the six models where the average curvature is calculated using Eq. (5.3). The results (Table 5.5) show the curvature values are smaller than the curvature values of *adv.PGD*, which indicates the curvature value can be effectively restrained by our proposed regularization.

	Capacity	*adv.PGD*			***adv.FGSMR***		
Models	(Million)	*PGD-inf*	*FGSM*	Average Curvature	*PGD-inf*	*FGSM*	Average Curvature
ResNet-18	11	0.444	0.506	0.487	0.458	0.51	0.324
ResNet-34	21	0.469	0.511	0.442	0.475	0.525	0.309
ResNet-50	23	0.448	0.512	0.565	0.479	0.528	0.338
WResNet-22x1	0.27	0.383	0.407	0.282	0.408	0.438	0.245
WResNet-22x5	6	0.438	0.495	0.502	0.462	0.495	0.262
WResNet-22x10	26	0.440	0.498	0.504	0.477	0.515	0.319

Table 5.5: Effect of network depth and width. The perturbed-data accuracy under *PGD-inf* and *FGSM* attacks are shown for robust models with different capacities. For network depth, *ResNet-18/34/50* with increasing depth is reported. For network width, *Wide ResNet-22×1/5/10×0×10* with increasing width are reported. As compared, Robust models achieved by *adv.PGD* are tested too. Capacity denotes the number of trainable parameters in the model.

5.4.4 Performance under Black-box Attack

In this section, we evaluate our proposed method based on transferable adversarial attack [LCLS17]. Following the transferable adversarial attack, three no-defense models and two robust models are taken as the source model, and six models that are trained by *Vanilla train*, *adv.FGSMR* ($\epsilon = 8.0/255$) and *adv.PGD* ($\epsilon = 8.0/255$) methods respectively are taken as target models. The adversarial examples under *PGD-inf* attack with ($\epsilon = 8.0/255$) are generated from the source model to attack the target model. The results (Table 5.6) show that models trained by *adv.FGSMR* achieve slightly higher perturbed-data accuracy than models trained by *adv.PGD* under transferable adversarial examples generated from both robust and non-defense models, which indicates *adv.FGSMR* can defend black-box attack as effectively as *adv.PGD*. We also observe that the perturbed-data accuracy achieved by *adv.FGSMR* is much closer to *adv.PGD* than *Vanilla train*, which indicates that *adv.FGSMR* learns a similar feature with *adv.PGD* but a different feature with *Vanilla train*.

Target model	*Vanilla train*		*adv.PGD*		***adv.FGSMR***	
Source model	*ResNet-18*	*ResNet-34*	*ResNet-18*	*ResNet-34*	*ResNet-18*	*ResNet-34*
Vanilla train(*ResNet-18*)	0.00	0.040	0.736	0.759	0.771	0.764
Vanilla train(*ResNet-34*)	0.070	0.016	0.735	0.758	0.772	0.764
Vanilla train(*ResNet-50*)	0.071	0.084	0.747	0.760	0.774	0.766
adv.PGD(*ResNet-18*)	0.792	0.787	0.444	0.584	0.606	0.614
adv.PGD(*ResNet-34*)	0.738	0.741	0.554	0.469	0.577	0.582

Table 5.6: The perturbed-data accuracy under transferable adversarial attack. The rows denote the three vanilla-trained models and two robust models which are used for generating transferable adversarial examples on the CIFAR-10 test set. The columns denote models trained by *Vanilla train*, *adv.FGSMR* and *adv.PGD* respectively that are used for testing.

5.5 Conclusion

In this chapter, we aimed to improve the adversarial robustness of the *adv.FGSM* method by adding a curvature regularization. We first identified the reason for the performance gap between *adv.FGSM* and *adv.PGD* by showing that the large difference in perturbed directions, caused by the increasing curvature along the FGSM perturbed direction, leads to a difference in adversarial robustness. To address this issue, we proposed the *adv.FGSMR* method, which adds a curvature regularization to restrain the growth of curvature along the FGSM perturbed direction. The evaluation of *adv.FGSMR* showed that it achieved comparable convergence speed on perturbed-data accuracy and took half the time to train one epoch compared to adv.PGD ($k = 20$). Under white-box attack, *adv.FGSMR* performed similarly to adv.PGD on the MNIST dataset and outperformed it on the CIFAR-10 dataset. Furthermore, it effectively defended transferable adversarial attacks, performing as well as *adv.PGD* under black-box attack.

Chapter 6

On Generalization of Graph Autoencoders with Adversarial Training

Adversarial training is an approach for increasing a model's resilience against adversarial perturbations. Such approaches have been demonstrated to result in models with feature representations that generalize better. However, limited works have been done on adversarial training of models on graph data. In this chapter, we raise such a question – does adversarial training improve the generalization of graph representations. We formulate L_2 and L_∞ versions of adversarial training in two powerful node embedding methods: graph autoencoder (GAE) and variational graph autoencoder (VGAE). We conduct extensive experiments on three main applications, i.e. link prediction, node clustering, graph anomaly detection of GAE and VGAE, and demonstrate that both L_2 and L_∞ adversarial training boost the generalization of GAE and VGAE.

6.1 Introduction

Networks are ubiquitous in plenty of real-world applications and they contain relationships between entities and attributes of entities. Modeling such data is challenging due to its non-Euclidean characteristic. Recently, graph embedding that converts graph data into low dimensional feature space has emerged as a popular method to model graph data, For example, DeepWalk [PARS14], node2vec [GL16] and LINE [TQW+15] learn graph embedding by extracting patterns from the graph. Graph Convolutions Networks (GCNs) [KW16a] learn graph embedding by repeated multiplication of normalized adjacency matrix and feature matrix. In particular, graph autoencoder (GAE) [KW16b, TGC+14, WCZ16] and variational graph autoencoder (VGAE) [KW16b] have been shown to be powerful node embedding methods as unsupervised learning. They have been applied to many machine learning tasks, e.g. node clustering [SHV20, TGC+14, SFK20], link prediction [SKB+18, KW16b], graph anomaly detection [PHvIP20, DLBL19] and etc.

Adversarial training is an approach for increasing a model's resilience against adversarial perturbations by including adversarial examples in the training set [MMS+17]. Several recent studies demonstrate that adversarial training improves feature representations leading to better performance for downstream tasks [UKE+20, SIE+20]. However, little work in this direction has been done for GAE and VGAE. Besides, real-world graphs are usually highly noisy and incomplete, which may lead to sub-optimal results for standard trained models [YZJ+19]. Therefore, we are interested to seek answers to the following two questions:

- *Does adversarial training improve generalization, i.e. the performance in applications of node embeddings learned by GAE and VGAE?*

- *Which factors influence this improvement?*

In order to answer the first question above, we first formulate L_2 and L_∞ adversarial training for GAE and VGAE. Then, we select three main tasks of VGAE and GAE: link prediction, node clustering, and graph anomaly detection for evaluating the generalization performance brought by adversarial training. Besides, we empirically explore which factors affect the generalization performance brought by adversarial training.

Contributions: To the best of our knowledge, we are the first to explore generalization for GAE and VGAE using adversarial training. We formulate L_2 and L_∞ adversarial training, and empirically demonstrate that both L_2 and L_∞ adversarial training boost the generalization with a large margin for the node embeddings learned by GAE and VGAE. An intriguing discovery is that the proposed adversarial training's generalization performance is more vulnerable to attribute perturbations than adjacency matrix perturbations and is insensitive to node degrees.

6.2 Related Work

Adversarial training has been extensively studied in images. It has been an important issue to explore whether adversarial training can help generalization. Tsipras et al. [TSE+18b] illustrates that adversarial robustness could conflict with a model's generalization by a designed simple task. However, Stutz et al. [SHS19] demonstrate that adversarial training with on-manifold adversarial examples helps the generalization. Besides, Salman et al. [SIE+20] and Utrera et al. [UKE+20] show that the latent features learned by adversarial training are improved and boost the performance of their downstream tasks.

Recently, a few works bring adversarial training to the graph data. Deng, Dong, and Zhu [DDZ19] and Sun et al. [SLGZ19] propose virtual graph adversarial training to promote the smoothness of a model. Feng et al. [FHTC19] propose graph adversarial training by inducing dynamical regularization. Dai et al. [DSZ+19] formulate an interpretable adversarial training for DeepWalk. Jin and Zhang [JZ19] introduce latent adversarial training for GCN, which train GCN based on the adversarial perturbed output of the first layer. Besides, several studies explored adversarial training based on adversarial perturbed edges for graph data [XCL+19, CWLX19, WLH19]. Among these works, part of the studies pays attention to achieving adversarial robustness while ignoring the effect of generalization [XCL+19, JZ19, CWLX19, WLH19, DSZ+19] and the others simply utilize perturbations on nodal attributes while not explore the effect of the perturbation on edges [DDZ19, SLGZ19, FHTC19]. The difference between these works and ours is two-fold: (1) We extend both L_∞ and L_2 adversarial training for graph models while the previous studies only explore L_2 adversarial training. (2) We focus on the generalization effect brought by adversarial training for unsupervised deep learning graph models, i.e. GAE and VGAE while most of the previous studies focus on adversarial robustness for supervised/semi-supervised models.

6.3 Preliminaries

We first summarize some notations and definitions used in this chapter. Following the commonly used notations, we use bold uppercase characters for matrices, e.g. $\boldsymbol{X}$, bold lowercase characters for vectors, e.g. $\boldsymbol{b}$, and normal lowercase characters for scalars, e.g. c. The i^{th} row of a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}_{i,:}$ and $(i,j)^{th}$ element of matrix $\boldsymbol{A}$ is denoted as $\boldsymbol{A}_{i,j}$. The i^{th} row of a matrix $\boldsymbol{X}$ is denoted by $\boldsymbol{x}_i$. We use **KL** for Kullback-Leibler divergence.

We consider an attributed network $\mathcal{G} = \{V, E, \boldsymbol{X}\}$ with $|V| = n$ nodes, $|E| = m$ edges and $\boldsymbol{X}$ node attributed matrix. $\boldsymbol{A}$ is the binary adjacency matrix of $\mathcal{G}$.

6.3.1 Graph Autoencoders

Graph autoencoders are a kind of unsupervised learning model on graph-structure data [KW16b], which aim at learning low dimensional representations for each node by reconstructing inputs. It has been demonstrated to achieve competitive results in multiple tasks, e.g. link prediction [SHV20, KW16b, SLH+19], node clustering [SHV20, TGC+14, SFK20], graph anomaly detection [DLBL19, PHvIP20]. Generally, graph autoencoder consists of a graph convolutional network for the encoder and an inner product for decoder [KW16b]. Formally, it can be expressed as follows:

$$\boldsymbol{Z} = GCN(\boldsymbol{A}, \boldsymbol{X}) \tag{6.1}$$

$$\hat{\boldsymbol{A}} = \sigma(\boldsymbol{Z}\boldsymbol{Z}^T), \tag{6.2}$$

where σ is the sigmoid function, GCN is a graph convolutional network, $\boldsymbol{Z}$ is the learned low dimensional representations and $\hat{\boldsymbol{A}}$ is the reconstructed adjacency matrix.

During the training phase, the parameters will be updated by minimizing the reconstruction loss. Usually, the reconstruction loss is expressed as cross-entropy loss between $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$ [KW16b]:

$$\mathcal{L}^{ae} = -\frac{1}{n^2} \sum_{(i,j)\in V\times V} \Big[\boldsymbol{A}_{i,j} log \hat{\boldsymbol{A}}_{i,j} + (1 - \boldsymbol{A}_{i,j}) log(1 - \hat{\boldsymbol{A}}_{i,j}) \Big]. \tag{6.3}$$

6.3.2 Variational Graph Autoencoders

Kipf and Welling [KW16b] introduced variational graph autoencoder (VGAE) which is a probabilistic model. VGAE consists of an inference model and a generative model. In their approach, the inference model, i.e. corresponding to the encoder of VGAE, is expressed as follows:

$$q(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{A}) = \prod_{i=1}^{n} q(\boldsymbol{z}_i|\boldsymbol{X}, \boldsymbol{A}),\ with\ q(\boldsymbol{z}_i|\boldsymbol{X}, \boldsymbol{A}) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_i, diag(\boldsymbol{\sigma}_i^2)), \tag{6.4}$$

where $\boldsymbol{\mu_i}$ and $\boldsymbol{\sigma_i}$ are learned by a GNN respectively. That is, $\boldsymbol{\mu} = GCN_{\mu}(\boldsymbol{X}, \boldsymbol{A})$ and $log\boldsymbol{\sigma} = GCN_{\sigma}(\boldsymbol{X}, \boldsymbol{A})$, with $\boldsymbol{\mu}$ is the matrix of stacking vectors $\boldsymbol{\mu_i}$; likewise, $\boldsymbol{\sigma}$ is the matrix of stacking vectors σ_i.

The generative model, i.e. corresponding to the decoder of autoencoder, is designed as an inner product between latent variables $\boldsymbol{Z}$, which is formally expressed as follows:

$$p(\boldsymbol{A}|\boldsymbol{Z}) = \prod_{i=1}^{n}\prod_{j=1}^{n} p(\boldsymbol{A}_{i,j}|\boldsymbol{z}_i,\boldsymbol{z}_j),\ with\, p(\boldsymbol{A}_{i,j}=1|\boldsymbol{z}_i,\boldsymbol{z}_j) = \sigma(\boldsymbol{z}_i^T\boldsymbol{z}_j). \tag{6.5}$$

During the training phase, the parameters will be updated by minimizing the variational lower bound $\mathscr{L}^{vae}$:

$$\mathscr{L}^{vae} = \mathbf{E}_{q(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{A})}[logp(\boldsymbol{A}|\boldsymbol{Z})] - \mathbf{KL}(q(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{A})||p(\boldsymbol{Z})), \tag{6.6}$$

where a Gaussian prior is adopted for $p(\boldsymbol{Z}) = \prod_i p(\boldsymbol{z}_i) = \prod_i \mathcal{N}(\mathbf{z}_i|0,\mathbf{I})$.

6.3.3 Adversarial Training

By now, multiple variants of adversarial training have been proposed and most of them are built on supervised learning and Euclidean data, e.g. FGSM-adversarial training [GSS14b], PGD-adversarial training [MMS+17], Trades [ZYJ+19a], MART [WZY+20] and etc. Here we introduce Trades that will be extended to GAE and VGAE settings in Section 6.4. Trades [ZYJ+19a] separates loss function into two terms:1) Cross-Entropy Loss for achieving natural accuracy; 2) Kullback-Leibler divergence for achieving adversarial robustness. Formally, given inputs (X,Y), it can be expressed as follows [ZYJ+19a]:

$$\min_{\theta} \mathbf{E}_{(X,Y)}[L(f_\theta(X),Y) + \lambda\cdot\mathbf{KL}(P(Y|X')||P(Y|X))], \tag{6.7}$$

where f_θ is a supervised model, X' is the adversarial examples that maximize **KL** divergence and $P(Y|X)$ is the output probability after softmax. λ is a tunable hyperparameter and it controls the strength of the **KL** regularization term.

6.4 Graph Adversarial Training

In this section, we formulate L_2 and L_∞ adversarial training for GAE and VGAE respectively.

6.4.1 Adversarial Training in Graph Autoencoder

Considering that: (1) the inputs of GAE contain adjacency matrix and attributes, (2) the latent representation $\boldsymbol{Z}$ is expected to be invariant to the input perturbation, we reformulate the loss function in Eq. (6.3) as follows:

$$\min_{\theta} \mathscr{L}^{ae} + \lambda \cdot \mathbf{KL}(P(\boldsymbol{Z}|\boldsymbol{A'},\boldsymbol{X'})||P(\boldsymbol{Z}|\boldsymbol{A},\boldsymbol{X})) \tag{6.8}$$

$$\boldsymbol{X}' = arg \max_{\|\boldsymbol{X}'-\boldsymbol{X}\|\leq\epsilon} \mathscr{L}^{ae}(\boldsymbol{A},\boldsymbol{X}),\ \boldsymbol{A}' = arg \max_{\|\boldsymbol{A}'-\boldsymbol{A}\|\leq\epsilon} \mathscr{L}^{ae}(\boldsymbol{A},\boldsymbol{X}) \tag{6.9}$$

where $\boldsymbol{A}'$ is the adversarial perturbed adjacency matrix and $\boldsymbol{X}'$ is the adversarial perturbed attributes. Here the important question is how to generate the perturbed adjacency matrix $\boldsymbol{A}'$ and attributes $\boldsymbol{X}'$ in Eq. (6.9).

Attributes Perturbation $\boldsymbol{X}'$. We generate the perturbed X' by projection gradient descent (PGD) [MMS+17]. We denote total steps as T.

For $\boldsymbol{X}'$ bounded by L_2 norm ball, the perturbed data in t-th step $\boldsymbol{X}^t$ is expressed as follows:

$$\boldsymbol{X}^t = \prod_{\mathscr{B}(\boldsymbol{X},\epsilon\|X\|_2)} (\boldsymbol{X}^{t-1} + \alpha \cdot g \cdot \|X\|_2 / \|g\|_2) \tag{6.10}$$

$$g = \nabla_{\boldsymbol{X}^{t-1}} \mathscr{L}^{ae}(\boldsymbol{A},\boldsymbol{X}^{t-1}) \tag{6.11}$$

where $\prod$ is the projection operator and $\mathscr{B}(\boldsymbol{X},\epsilon\|X\|_2)$ is the L_2 norm ball of nodal attributes $\boldsymbol{x}_i : \{\boldsymbol{x}_i' : \|\boldsymbol{x}_i' - \boldsymbol{x}_i\|_2 \leq \epsilon\|\boldsymbol{x}_i\|_2\}$.

For $\boldsymbol{X}'$ bounded by L_∞ norm ball, the perturbed data in t-th step $\boldsymbol{X}^t$ is expressed as follows:

$$\boldsymbol{X}^t = \prod_{\mathscr{B}(\boldsymbol{X},\epsilon)} (\boldsymbol{X}^{t-1} + \alpha \cdot g) \tag{6.12}$$

$$g = sgn(\nabla_{\boldsymbol{X}^{t-1}} \mathscr{L}^{ae}(\boldsymbol{A},\boldsymbol{X}^{t-1})), \tag{6.13}$$

where $\mathscr{B}(\boldsymbol{X},\epsilon)$ is the L_∞ norm ball of nodal attributes $\boldsymbol{x}_i : \{\boldsymbol{x}_i' : \|\boldsymbol{x}_i' - \boldsymbol{x}_i\|_\infty \leq \epsilon\}$ and $sgn(\cdot)$ is the sign function.

Adjacency Matrix Perturbation $\boldsymbol{A}'$. Adjacency matrix perturbation includes two-fold: (1) perturb node connections, i.e. Adding or dropping edges, (2) perturb the strength of information flow between nodes, i.e. the strength of correlation between nodes. Here we choose to perturb the strength of information flow between nodes and leave the perturb of node connections for future work. Specifically, we add weight for each edge and change these weights in order to perturb the strength of the information flow. Formally, given the adjacency matrix A, the weighted adjacency matrix $\tilde{\boldsymbol{A}}$ is expressed as $\boldsymbol{A} \odot \boldsymbol{M}$ where the elements of $\boldsymbol{M}$ are continuous and its values are initialized as the same value as $\boldsymbol{A}$. $\odot$ denotes the element-wise product. Formally, $\boldsymbol{A}'$ is expressed as follows:

$$\boldsymbol{M}' = arg \max_{\|\boldsymbol{M}'-\boldsymbol{M}\| \leq \epsilon} \mathscr{L}^{ae}(\tilde{\boldsymbol{A}}, \boldsymbol{X}) \tag{6.14}$$

$$\boldsymbol{A}' = \boldsymbol{A} \odot \boldsymbol{M}'. \tag{6.15}$$

For $\boldsymbol{A}'$ bounded by L_2 norm ball, the perturbed data in t-th step $\boldsymbol{A}^t$ is expressed as follows:

$$g = \nabla_{\boldsymbol{M}^{t-1}} \mathscr{L}^{ae}(\tilde{\boldsymbol{A}}^{t-1}, \boldsymbol{X}) \tag{6.16}$$

$$\boldsymbol{M}^t = \prod_{\mathscr{B}(\boldsymbol{M}, \epsilon \|M\|_2)} (\boldsymbol{M}^{t-1} + \alpha \cdot g \cdot \|M\|_2 / \|g\|_2) \tag{6.17}$$

$$\boldsymbol{A}^t = \tilde{\boldsymbol{A}}^t = \boldsymbol{A} \odot \boldsymbol{M}^t. \tag{6.18}$$

For $\boldsymbol{A}'$ bounded by L_∞ norm ball, the perturbed data in t-th step $\boldsymbol{A}^t$ is expressed as follows:

$$g = sgn(\nabla_{\boldsymbol{M}^{t-1}} \mathscr{L}^{ae}(\tilde{\boldsymbol{A}}^{t-1}, \boldsymbol{X})) \tag{6.19}$$

$$\boldsymbol{M}^t = \prod_{\mathscr{B}(\boldsymbol{M}, \epsilon)} (\boldsymbol{M}^{t-1} + \alpha \cdot g) \tag{6.20}$$

$$\boldsymbol{A}^t = \tilde{\boldsymbol{A}}^t = \boldsymbol{A} \odot \boldsymbol{M}^t. \tag{6.21}$$

6.4.2 Adversarial Training in Variational Graph Autoencoder

Similarly to GAE, we reformulate the loss function for training VGAE (Eq. (6.6)) as follows:

$$\min_\theta \mathscr{L}^{vae} + \lambda \cdot \mathbf{KL}(P(\boldsymbol{Z}|\boldsymbol{A'}, \boldsymbol{X'}) || P(\boldsymbol{Z}|\boldsymbol{A}, \boldsymbol{X})) \tag{6.22}$$

$$\boldsymbol{X}' = arg \max_{\|\boldsymbol{X}'-\boldsymbol{X}\| \leq \epsilon} \mathscr{L}^{vae}(\boldsymbol{A}, \boldsymbol{X}), \ \boldsymbol{A}' = arg \max_{\|\boldsymbol{A}'-\boldsymbol{A}\| \leq \epsilon} \mathscr{L}^{vae}(\boldsymbol{A}, \boldsymbol{X}) \tag{6.23}$$

We generate $\boldsymbol{A}'$ and $\boldsymbol{X}'$ exactly the same way as with GAE (replacing $\mathscr{L}^{ae}$ with $\mathscr{L}^{vae}$ in Eq. (10-21).)

For convenience, we abbreviate L_2 and L_∞ adversarial training as AT-2 and AT-Linf respectively in the following tables and figures where L_2/L_∞ denote both attributes and adjacency matrix perturbation are bounded by L_2/L_∞ norm ball.

In practice, we train models by alternatively adding adjacency matrix perturbation and attributes perturbation [1].

6.5 Experiments

In this section, we present the results of the performance evaluation of L_2 and L_∞ adversarial training under three main applications of GAE and VGAE: link prediction, node clustering, and graph anomaly detection. Then we conduct parameter analysis experiments to explore which factors influence the performance.

Datasets. We used six real-world datasets: Cora, Citeseer, and PubMed for link prediction and node clustering tasks, and BlogCatalog, ACM, and Flickr for the graph anomaly detection task. The detailed descriptions of the six datasets are shown in Table 6.1.

Model Architecture. All our experiments are based on the GAE/VGAE model where the encoder/inference model is consisted of a two-layer GCN by default.

Table 6.1: Datasets Descriptions.

DataSets	Cora	Citeseer	PubMed	BlogCatalog	ACM	Flickr
#Nodes	2708	3327	19717	5196	16484	7575
#Links	5429	4732	44338	171743	71980	239738
#Features	1433	3703	500	8189	8337	12074

6.5.1 Link Prediction

Metrics. Following [KW16b], we use the area under a receiver operating characteristic curve (AUC) and average precision (AP) as the evaluation metric. We conduct 30 repeat experiments with random splitting datasets into 85%, 5%, and 10% for training sets, validation sets, and test sets respectively. We report the mean and standard deviation values on test sets.

Parameter Settings. We train models on Cora and Citeseer datasets with 600 epochs, and PubMed with 800 epochs. All models are optimized with Adam optimizer and 0.01 learning rate. The λ is set to 4. For attributes perturbation, the ϵ is set to 3e-1 and 1e-3 on Citeseer and Cora, 1 and 5e-3 on PubMed for L_2

[1] We find that optimizing models by alternatively adding these two perturbations is better than adding these two perturbations together.

and L_∞ adversarial training respectively. For adjacency matrix perturbation, the ϵ is set to 1e-3 and 1e-1 on Citeseer and Cora, and 1e-3 and 3e-1 on PubMed for L_2 and L_∞ adversarial training respectively. The steps T is set to 1. The α is set to $\frac{\epsilon}{T}$.

For standard training GAE and VGAE, we run the official Pytorch geometric code [2] with 600 epochs for Citeseer and Cora datasets, 1000 epochs [3] for PubMed dataset. Other parameters are set the same as in [KW16b].

Experimental Results. The results are shown in Table 6.2. It can be seen that both L_2 and L_∞ Adversarial trained GAE and VGAE models consistently boost their performance for both AUC and AP metrics on Cora, Citeseer, and PubMed datasets. Specifically, the improvements on the Cora and Citeseer datasets reach at least 2% for both GAE and VGAE (Table 6.2). The improvements on PubMed are relatively small with around 0.3%.

Table 6.2: Results for Link Prediction.

Methods	Cora		Citeseer		PubMed	
	AUC (in%)	AP (in%)	AUC (in%)	AP (in%)	AUC (in%)	AP (in%)
GAE	90.6 ± 0.9	91.2 ± 1.0	88.0 ± 1.2	89.2 ± 1.0	96.8 ± 0.2	97.1 ± 0.2
AT-L2-GAE	**93.0** ± 0.9	**93.5** ± 0.6	**92.5** ± 0.7	**93.2** ± 0.6	**97.2** ± 0.2	**97.4** ± 0.2
AT-Linf-GAE	92.8 ± 1.1	93.4 ± 1.0	92.3 ± 0.9	92.6 ± 1.1	96.9 ± 0.2	97.3 ± 0.2
VGAE	89.8 ± 0.9	90.3 ± 1.0	86.6 ± 1.4	87.6 ± 1.3	96.2 ± 0.4	96.3 ± 0.4
AT-L2-VGAE	**92.8** ± 0.6	**93.1** ± 0.6	90.7 ± 1.1	91.1 ± 0.9	**96.6** ± 0.2	**96.7** ± 0.2
AT-Linf-VGAE	92.2 ± 1.2	92.3 ± 1.3	**91.9** ± 0.8	**92.0** ± 0.6	96.5 ± 0.2	96.6 ± 0.3

6.5.2 Node Clustering

Metrics. Following [PHL$^+$18, XPDY14], we use accuracy (ACC), normalized mutual information (NMI), precision, F-score(F1), and average rand index (ARI) as our evaluation metrics. We conduct 10 repeat experiments. For each experiment, datasets are randomly split into training sets(85% edges), validation sets (5% edges), and test sets (10% edges). We report the mean and standard deviation values on test sets.

Parameter Settings. We train GAE models on Cora and Citeseer datasets with 400 epochs, and PubMed dataset with 800 epochs. We train VGAE models on the Cora and Citeseer datasets with 600 epochs and the PubMed dataset with 800

[2] `https://github.com/rusty1s/pytorch_geometric/blob/master/examples/autoencoder.py`

[3] Considering PubMed is big graph data, we use more epochs in order to avoid underfitting.

epochs. All models are optimized by Adam optimizer with 0.01 learning rate. The λ is set to 4. For attributes perturbation, the ϵ is set to 5e-1 and 1e-3 on the both Cora and Citeseer dataset, and 1 and 5e-3 on the PubMed dataset for L_2 and L_∞ adversarial training respectively. For adjacency matrix perturbation, the ϵ is set to 1e-3 and 1e-1 on Cora and CiteSeer, 1e-3 and 3e-1 on PubMed for L_2 and L_∞ adversarial training respectively. The steps T is set to 1. The α is set to $\frac{\epsilon}{T}$.

Likewise, for standard GAE and VGAE, we run the official Pytorch geometric code with 400 epochs for the Citeseer and Cora datasets, and 800 epochs for the PubMed dataset.

Experimental Results. The results are showed in Table 6.3, Table 6.4 and Table 6.5. It can be seen that both L_2 and L_∞ adversarially trained models consistently outperform the standard trained models for all metrics. In particular, on Cora and Citeseer datasets, both L_2 and L_∞ adversarial training improve the performance with a large margin for all metrics, i.e. at least +5.4% for GAE, +6.7% for VGAE on Cora dataset (Table 6.3), and at least +5.8% for GAE, +5.6% for VGAE on Citeseer dataset (Table 6.4).

Table 6.3: Results for Node Clustering on Cora.

Methods	Acc (in%)	NMI (in%)	F1 (in%)	Precision (in%)	ARI (in%)
GAE	61.6 ± 3.4	44.9 ± 2.3	60.8 ± 3.4	62.5 ± 3.5	37.2 ± 3.2
AT-L2-GAE	67.0 ± 3.0	50.8 ± 1.7	66.6 ± 1.7	69.4 ± 1.7	**44.1** ± 4.1
AT-Linf-GAE	**67.1** ± 3.8	**51.4** ± 1.9	**67.5** ± 2.8	**70.7** ± 2.2	43.4 ± 4.3
VGAE	58.7 ± 2.7	42.3 ± 2.2	57.3 ± 3.2	58.8 ± 3.5	34.6 ± 2.8
AT-L2-VGAE	**67.3** ± 3.8	**50.5** ± 2.1	**66.1** ± 4.1	**67.5** ± 3.8	**44.3** ± 3.3
AT-Linf-VGAE	65.4 ± 2.3	49.5 ± 1.6	64.0 ± 2.3	65.8 ± 3.0	42.9 ± 2.8

Table 6.4: Results for Node Clustering on Citeseer.

Methods	Acc (in%)	NMI (in%)	F1 (in%)	Precision (in%)	ARI (in%)
GAE	51.8 ± 2.6	28.0 ± 1.9	50.6 ± 3.1	55.1 ± 3.1	22.8 ± 2.3
AT-L2-GAE	**61.6** ± 2.3	36.3 ± 1.4	**58.8** ± 2.1	60.9 ± 1.4	**34.6** ± 2.3
AT-Linf-GAE	60.2 ± 2.8	**38.0** ± 2.3	57.0 ± 2.7	**61.1** ± 1.6	34.1 ± 3.4
VGAE	53.6 ± 3.5	28.4 ± 3.3	51.1 ± 3.8	53.2 ± 4.1	26.1 ± 3.5
AT-L2-VGAE	59.2 ± 2.3	35.1 ± 2.3	57.3 ± 2.3	60.4 ± 3.1	33.0 ± 2.4
AT-Linf-VGAE	**60.4** ± 1.5	**36.5** ± 1.4	**58.2** ± 1.4	**61.1** ± 1.4	**34.7** ± 2.0

Table 6.5: Results for Node Clustering on PubMed.

Methods	Acc (in%)	NMI (in%)	F1 (in%)	Precision (in%)	ARI (in%)
GAE	66.2 ± 2.0	27.9 ± 3.7	65.0 ± 2.3	68.8 ± 2.2	27.1 ± 3.3
AT-L2-GAE	67.5 ± 2.9	30.4 ± 5	66.7 ± 3.3	70.2 ± 3.1	28.9 ± 4.8
AT-Linf-GAE	**68.4** ± 1.6	**31.9** ± 3.2	**67.7** ± 1.9	**70.9** ± 1.8	**30.2** ± 2.8
VGAE	67.5 ± 2.0	29.4 ± 3.2	66.5 ± 2.2	69.9 ± 2.2	28.4 ± 3.2
AT-L2-VGAE	**69.8** ± 2.0	**33.2** ± 3.4	**69.4** ± 2.3	**71.7** ± 2.5	**32.5** ± 3.2
AT-Linf-VGAE	68.5 ± 1.2	30.7 ± 2.5	67.4 ± 1.5	70.1 ± 1.5	30.4 ± 2.0

6.5.3 Graph Anomaly Detection

We strictly follow the experimental protocol outlined in [DLBL19] for graph anomaly detection. In [DLBL19], the authors take reconstruction errors of attributes and links as the anomaly scores. Specifically, the node with larger scores is more likely to be considered anomalies.

Model Architecture. Different from link prediction and node clustering, the model architecture in graph anomaly detection not only contains a structure reconstruction decoder, i.e. link reconstruction, but also contains an attribute reconstruction decoder. We adopt the same model architecture as in the official code of [DLBL19] where the encoder consists of two GCN layers and the decoder of structure reconstruction decoder consists of a GCN layer and an InnerProduction layer, and the decoder of attributes reconstruction decoder consists of two GCN layers.

Metrics. Following [DLBL19, PHvIP20], we use the area under the receiver operating characteristic curve (ROC-AUC) as the evaluation metric.

Parameter Settings. We set the α in anomaly scores to 0.5 where it balances the structure reconstruction errors and attributes reconstruction errors. We train the GAE model on Flickr, BlogCatalog, and ACM datasets with 300 epochs. We set λ to 5. For adjacency matrix perturbation, we set ϵ to 3e-1, 5e-5 on both BlogCatalog and ACM datasets, 1e-3 and 1e-6 on Flickr dataset for L_∞ and L_2 adversarial training respectively. For attributes perturbations, we set ϵ to 1e-3 on BlogCatalog for both L_∞ and L_2 adversarial training, 1e-3 and 1e-2 on ACM for L_∞ and L_2 adversarial training respectively, 5e-1 and 3e-1 on Flickr for L_∞ and L_2 adversarial training respectively. We set steps T to 1 and the α to $\frac{\epsilon}{T}$

Anomaly Generation. Following [DLBL19], we inject two kinds of anomaly by perturbing structure and nodal attributes respectively:

- Structure anomalies. We randomly select s nodes from the network and then make those nodes fully connected, and then all the s nodes forming the clique are labeled as anomalies. t cliques are generated repeatedly and totally there are $s \times t$ structural anomalies.

- Attribute anomalies. We first randomly select $s \times t$ nodes as the attribute perturbation candidates. For each selected node v_i, we randomly select another k node from the network and calculate the Euclidean distance between v_i and all the k nodes. Then the node with the largest distance is selected as v_j and the attributes of node v_j are changed to the attributes of v_i.

In this experiment, we set $s = 15$ and $t = 10, 15, 20$ for BlogCatalog, Flickr, and ACM respectively which are the same to [DLBL19, PHvIP20].

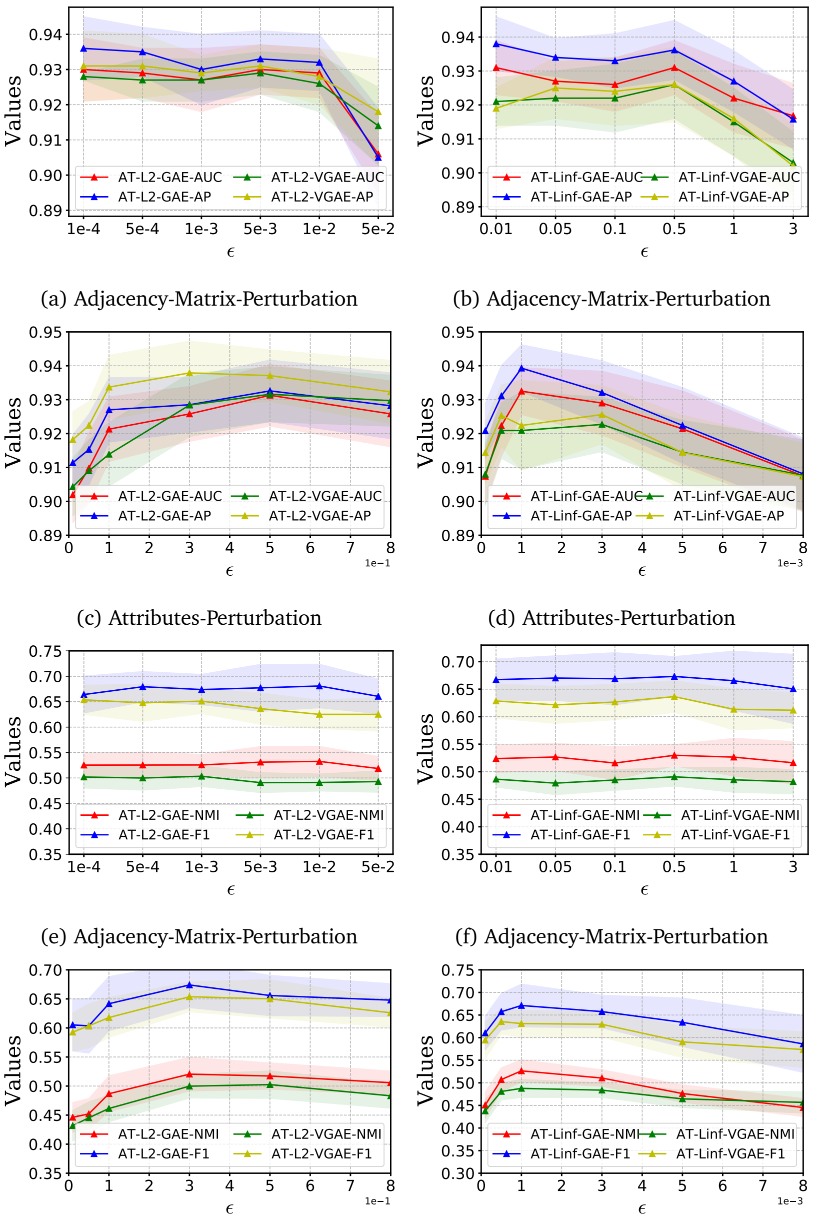
Experimental Results. From Table 6.6, it can be seen that both L_2 and L_∞ adversarial training boost the performance in detecting anomalous nodes. Since adversarial training tends to learn feature representations that are less sensitive to perturbations in the inputs, we conjecture that the adversarially trained node embeddings are less influenced by the anomalous nodes, which helps the graph anomaly detection. A similar claim is also made in the image domain [SAP+20] where they demonstrate adversarial training of autoencoders is beneficial to novelty detection.

Table 6.6: Results w.r.t. AUC (in%) for Graph Anomaly Detection.

Methods	Flickr	BlogCatalog	ACM
GAE	80.2 ± 1.3	82.9 ± 0.3	72.5 ± 0.6
AT-L2-GAE	$\mathbf{84.9} \pm 0.2$	$\mathbf{84.7} \pm 1.4$	74.2 ± 1.7
AT-Linf-GAE	81.1 ± 1.1	82.8 ± 1.3	$\mathbf{75.3} \pm 0.9$

6.6 Understanding Adversarial Training

In this section, we explore the impact of three hyper-parameters on the performance of GAE and VGAE with adversarial training, i.e. the ϵ, λ and T in generating $\boldsymbol{A}'$ and $\boldsymbol{X}'$. These three hyper-parameters are commonly considered to control the strength of regularization for adversarial robustness [ZYJ+19a]. Besides, we explore the relationship between the improvements achieved by adversarial training and node degree.



(a) Adjacency-Matrix-Perturbation

(b) Adjacency-Matrix-Perturbation

(c) Attributes-Perturbation

(d) Attributes-Perturbation

(e) Adjacency-Matrix-Perturbation

(f) Adjacency-Matrix-Perturbation

(g) Attributes-Perturbation

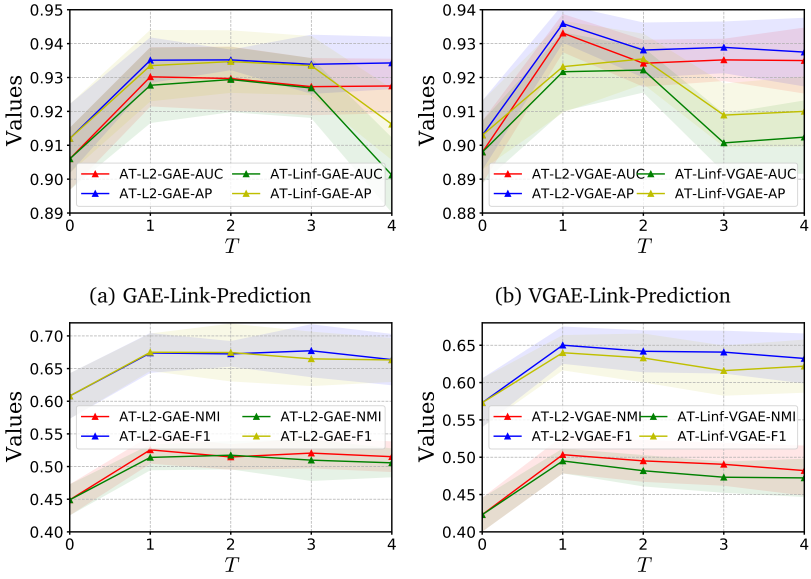
(h) Attributes-Perturbation

Figure 6.1: The impact of ϵ in adjacency matrix perturbation and attributes perturbation. (a)-(d) shows AUC/AP values for the link prediction task and (e)-(h) shows NMI/F1 values for the node clustering task. Dots denote mean values with 30 repeated runs.

6.6.1 The Impact of ϵ

The experiments are conducted on link prediction and node clustering tasks based on the Cora dataset. We fix ϵ to 5e-1 and 1e-3 on adjacency matrix perturbation for L_∞ and L_2 adversarial training respectively when varying ϵ on attributes perturbation. We fix ϵ to 1e-3 and 3e-1 on attributes perturbation for L_∞ and L_2 adversarial training respectively when varying ϵ on adjacency matrix perturbation.

The results are shown in Fig. 6.1. From Fig. 6.1, we can see that the performance is less sensitive to adjacency matrix perturbation and more sensitive to attributes perturbation. Besides, it can be seen that there is an increase and then a decreasing trend when increasing ϵ for attributes perturbation. We conjecture that it is because too large perturbations on attributes may destroy useful information in attributes. Therefore, it is necessary to carefully adapt the perturbation magnitude ϵ when we apply adversarial training for improving the generalization of a model.



(a) GAE-Link-Prediction

(b) VGAE-Link-Prediction

(c) GAE-Node-Clustering

(d) VGAE-Node-Clustering

Figure 6.2: The impact of steps T. Dots denote mean AUC/AP values for the link prediction task and mean NMI/F1 values for the node clustering task.

6.6.2 The Impact of T

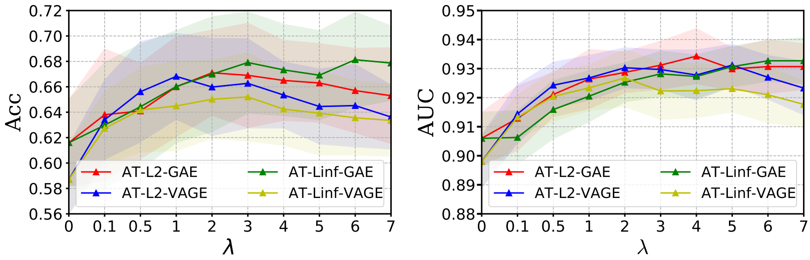
The experiments are conducted on link prediction and node clustering tasks based on the Cora dataset. For L_2 adversarial training, we set ϵ to 1e-3 and 5e-1 for adjacency matrix perturbation and attributes perturbation respectively. For L_∞ adversarial training, we set ϵ to 1e-1 and 1e-3 for adjacency matrix perturbation and attributes perturbation respectively. We set λ to 4.

Results are shown in Fig. 6.2. From Fig. 6.2, we can see that there is a slight drop in both link prediction and node clustering tasks when increasing T from 2 to 4, which implies that a big T is not helpful to improve the generalization of node embeddings learned by GAE and VGAE. We suggest that one step is a good choice for generating adjacency matrix perturbation and attributes perturbation in both L_2 and L_∞ adversarial training.

6.6.3 The Impact of λ

The experiments are conducted on link prediction and node clustering tasks based on the Cora dataset. Likewise, for L_2 adversarial training, ϵ is set to 1e-3 and 5e-1 for adjacency matrix perturbation and attributes perturbation respectively. For L_∞ adversarial training, ϵ is set to 1e-1 and 1e-3 for adjacency matrix perturbation and attributes perturbation respectively. T is set to 1.

Results are shown in Fig. 6.3. From Fig. 6.3, it can be seen that there is a significant increasing trend with the increase of λ, which indicates the effectiveness of both L_2 and L_∞ adversarial training in improving the generalization of GAE and VGAE. Besides, we also notice that a too-large λ is not necessary and may lead to a negative effect in the generalization of GAE and VGAE.



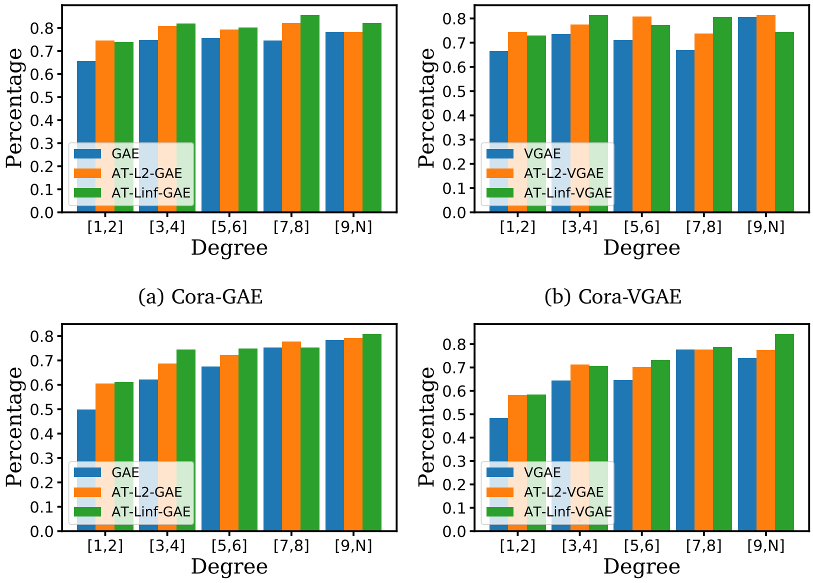
(a) Node Clustering (b) Link Prediction

Figure 6.3: The impact of λ. λ is varied from 0 to 7. Dots denote the mean Acc for the node clustering task and the mean AUC for the link prediction task. Experiments are conducted with 30 repeated runs.

6.6.4 Performance w.r.t. Node Degree

In this section, we explore whether the performance of adversarial trained GAE/VGAE is sensitive to the degree of nodes. To conduct this experiment, we first learn node embeddings from Cora and Citeseer datasets by GAE/VGAE with L_2/L_∞ adversarial training and standard training respectively. The hyperparameters are set the same as in the Node clustering task. Then we build a linear classification based on the learned node embeddings. The accuracy with respect to the node degree distribution is shown in Fig. 6.4.

From Fig. 6.4, it can be seen seem that for most degree groups, both L_2 and L_∞ adversarially trained models outperform standard trained models, which indicates that both L_2 and L_∞ adversarial training improve the generalization of GAE and VGAE with different degrees. However, we also notice that adversarial training does not achieve a significant improvement in [9,N] group. We conjecture that it is because node embeddings with very large degrees already achieve a high generalization.



(a) Cora-GAE

(b) Cora-VGAE

(c) Citeseer-GAE

(d) Citeseer-VGAE

Figure 6.4: Performance of GAE/VGAE and adversarial trained GAE/VGAE w.r.t. node degrees in Cora and Citeseer networks.

6.7 Conclusion

In this chapter, we first formulated L_2 and L_∞ adversarial training for GAE and VGAE, and then presented their impact on the generalization performance. The extensive experiments showed that both L_2 and L_∞ adversarial trained GAE and VGAE outperform GAE and VGAE with standard training, indicating that both L_2 and L_∞ adversarial training improve the generalization of GAE and VGAE. Besides, we also found that the generalization performance achieved by the L_2 and L_∞ adversarial training is more sensitive to attributes perturbation than adjacency matrix perturbation, and not sensitive to node degree. In addition, the parameter analysis suggested that a too large λ, ϵ, and T would lead to a negative effect on the performance w.r.t. generalization.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we focused on adversarial examples and explore their roles in building robust models. Concretely, we have investigated the following research questions.

How to improve the transferability of adversarial examples? (RQ1). In Chapter 2, we presented our study of the transferability of adversarial examples. We experimentally showed that adversarial examples generated by the existing standard adversarial attacks such as the PGD, FGSM, and C&W attacks, have low transferability. That is, the adversarial examples generated by the white-box model are hard to attack other models successfully. Furthermore, we showed that the transferability can be greatly improved by stabilizing the attack directions, resulting in a practical and strong black-box attack. In experiments, we demonstrated that the black-box models including robust-trained models are not robust to the proposed DA attack.

Can we effectively address the three challenges: (1) Robust overfitting, (2) Trade-off between clean accuracy and robust accuracy, (3) High training cost? (RQ2). Despite adversarial training (AT) has been demonstrated effective to improve adversarial robustness. However, there are still three challenging issues when we apply AT for improving adversarial robustness: ❶ Trade-off between clean accuracy and robust accuracy, ❷ Robust overfitting, ❸ Expensive training cost. We proposed methods to mitigate these issues.

- **CAT**. In Chapter 3, we presented our theoretical analysis for AT. We found that adversarial examples generated by maximizing robust error could lead to an inconsistency between adversarial examples and its label. We further showed that the proposed calibrated adversarial examples reduce the semantic changes on the input and keep the consistency between adversarial examples and the label. Based on the calibrated adversarial examples, we proposed Calibrated Adversarial Training (CAT) that boosts adversarial robustness while keeping a good trade-off between clean accuracy and robust accuracy, indicating that there could be strong connectivity between the trade-off and the consistency of labels and adversarial examples.

- **WOT**. In Chapter 4, We drew a connection between robust overfitting and optimization trajectories. We found that refining the optimization trajectories by maximizing the robust performance on unseen data effectively improves adversarial robustness while it has negligible or even no sacrifice on clean accuracy. In experiments, we showed that refining optimization trajectories can find flatter minima. Overall, our results suggested that optimization trajectories play a key role in mitigating robust overfitting and improving adversarial robustness.

- **Bridging FGSM-AT and PGD-AT**. In Chapter 5, we bridged FGSM-AT and PGD-AT by introducing a curvature regularization. We found that reducing the difference between adversarial examples from the FGSM and PGD attacks can avoid the “catastrophic overfitting” issue of FGSM-AT. In experiments, we showed that the proposed curvature regularization can decrease the iterations of the inner loop to 1 for PGD-AT and obtain a 2x time speed up while keeping the same adversarial robustness.

Can adversarial examples enhance the representation learning of graph neural networks? (RQ3). In Chapter 6, we explored the impact of adversarial examples on the generalization of graph autoencoder (GAE) and variational graph autoencoder (VGAE). We demonstrated that, with optimal choice of the magnitude of adversarial perturbations and the strength of the regularization based on adversarial examples, AT can boost the generalization with a large margin for the node embedding learned by GAE/ VGAE. This study indicates that there is a larger-than-expected space for applying adversarial examples for boosting task performance such as feature representation except for robust performance.

7.2 Future Work

Many studies indicate that current deep networks lack the generalization abilities of human perception, being susceptible to small input variations, viewpoint changes, and occlusions. Despite various attempts to enhance the robustness of these models, it remains an ongoing challenge to match the robustness of human perception. In this section, we outline some of the most promising directions for future development.

Offline AT. While PGD-AT achieves promising performance in improving adversarial robustness, the online adversarial examples generation occupies much computing resources. Therefore, developing offline AT is a promising future work where adversarial examples are collected offline. Since adversarial examples can be obtained with high transferability [DPSZ19, XZZ+19, Nes83], it might be also possible to generate static transferable adversarial examples that can attack models across the training period. Adversarial robustness might be achieved by training a model on these static transferable adversarial examples.

Optimal Adversarial Examples for AT. Although many variants of AT have been proposed, optimal adversarial examples for AT are still unknown. Several studies [ZXH+20a, SCW20, BGH19] experimentally showed dynamically changing ϵ can mitigate the trade-off between adversarial robustness and clean accuracy. However, most of them are heuristically proposed and lack a theoretical analysis and guarantee. Therefore, a theoretical analysis for optimal adversarial examples is critical for understanding AT and achieving a better trade-off between adversarial robustness and clean accuracy.

Universal Robustness. There are many different types of robustness. For an example, robustness to common corruptions [HD19], robustness to pixel-based adversarial attacks [GSS14a, MMS+18], robustness to spatial-based adversarial attacks [TB19, KDS19]. Although there are various methods proposed for improving this robustness respectively [MMS+18, HMC+19, KDS19], there are rare studies to boost this robustness altogether. [LCO19] show that the pixel-based adversarial robustness is not correlated with robustness to common corruptions. [SLL21] point out that increasing the robustness of pixel-based adversarial attacks leads to a decreased robustness of rotation-based adversarial attacks. Developing a general framework to improve universal robustness that can robust all these perturbations or attacks is important.

Bibliography

[ACFH20] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020. (Cited on pages 4, 52, 54, and 68.)

[ACW18] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018. (Cited on pages 5, 42, 54, 55, 70, and 78.)

[AEIK18] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018. (Cited on page 4.)

[AF20] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. In *NIPS*, 2020. (Cited on pages 6 and 44.)

[AM18] Naveed Akhtar and Ajmal Mian. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey, 2018. (Cited on page 78.)

[AUH$^+$19] Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? *Advances in Neural Information Processing Systems*, 32, 2019. (Cited on pages 5 and 62.)

[BGH19] Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*, 2019. (Cited on pages 5, 44, and 111.)

[BJM06] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. (Cited on page 49.)

[BZJ$^+$20] Yang Bai, Yuyuan Zeng, Yong Jiang, Yisen Wang, Shu-Tao Xia, and Weiwei Guo. Improving query efficiency of black-box adversarial attack. In *European Conference on Computer Vision*, pages 101–116. Springer, 2020. (Cited on page 4.)

[CDLS18] Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018. (Cited on pages 44 and 45.)

[CH20a] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020. (Cited on page 68.)

[CH20b] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020. (Cited on pages 2, 3, 5, and 68.)

[CLC$^+$20] Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. Cat: Customized adversarial training for improved robustness. *arXiv preprint arXiv:2002.06789*, 2020. (Cited on page 44.)

[CRK19] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings* *of Machine Learning Research*, pages 1310–1320. PMLR, 09–15 Jun 2019. (Cited on pages 14, 18, and 42.)

[CRS+19] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019. (Cited on pages 5 and 62.)

[CW17a] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 3–14, 2017. (Cited on page 1.)

[CW17b] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. (Cited on pages 2, 3, 14, 17, 45, 52, 68, and 78.)

[CWLX19] Jinyin Chen, Yangyang Wu, Xiang Lin, and Qi Xuan. Can adversarial network attack be defended? *arXiv preprint arXiv:1903.05994*, 2019. (Cited on page 93.)

[CZL+20] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2020. (Cited on pages 6, 62, 63, and 67.)

[CZS+17] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017. (Cited on pages 4 and 78.)

[CZW+22] Tianlong Chen, Zhenyu Zhang, Pengjun Wang, Santosh Balachandra, Haoyu Ma, Zehao Wang, and Zhangyang Wang. Sparsity winning twice: Better robust generaliztion from more efficient training. *arXiv preprint arXiv:2202.09844*, 2022. (Cited on pages 6 and 63.)

[DCP+21] Yinpeng Dong, Shuyu Cheng, Tianyu Pang, Hang Su, and Jun Zhu. Query-efficient black-box adversarial attacks guided by a transfer-based prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. (Cited on page 4.)

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. (Cited on page 66.)

[DDZ19] Zhijie Deng, Yinpeng Dong, and Jun Zhu. Batch virtual adversarial training for graph convolutional networks. *arXiv preprint arXiv:1902.09192*, 2019. (Cited on page 93.)

[DLBL19] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 594–602. SIAM, 2019. (Cited on pages 92, 94, 101, and 102.)

[DLP+18] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. (Cited on pages 14, 17, 19, 20, 25, and 27.)

[DPSZ19] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. (Cited on pages 4, 17, 20, 24, 27, and 111.)

[DR17] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017. (Cited on pages 6 and 64.)

[DSLH19] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. In *International Conference on Learning Representations*, 2019. (Cited on page 43.)

[DSZ+19] Quanyu Dai, Xiao Shen, Liang Zhang, Qiang Li, and Dan Wang. Adversarial training methods for network embedding. In *The*

World Wide Web Conference, pages 329–339, 2019. (Cited on page 93.)

[DXY$^+$21] Yinpeng Dong, Ke Xu, Xiao Yang, Tianyu Pang, Zhijie Deng, Hang Su, and Jun Zhu. Exploring memorization in adversarial training. *arXiv preprint arXiv:2106.01606*, 2021. (Cited on pages 6, 63, and 64.)

[EEF$^+$18] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. (Cited on pages 1 and 4.)

[EEPK05] Andre Elisseeff, Theodoros Evgeniou, Massimiliano Pontil, and Leslie Pack Kaelbing. Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6(1), 2005. (Cited on page 62.)

[EIA18] Logan Engstrom, Andrew Ilyas, and Anish Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018. (Cited on page 75.)

[EKM$^+$18] Gamaleldin F. Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. In *Advances in Neural Information Processing Systems*, volume 2018-December, pages 842–852. Neural information processing systems foundation, 2018. (Cited on page 16.)

[FHTC19] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering*, 2019. (Cited on page 93.)

[FKMN20] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020. (Cited on pages 6 and 64.)

[FO21] Chris Finlay and Adam M Oberman. Scaleable input gradient regularization for adversarial robustness. *Machine Learning with Applications*, 3:100017, 2021. (Cited on page 5.)

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014. (Cited on pages 14, 41, 62, and 78.)

[GFW18] Chuan Guo, Jared S Frank, and Kilian Q Weinberger. Low frequency adversarial perturbation. *arXiv preprint arXiv:1809.08758*, 2018. (Cited on page 42.)

[GL16] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016. (Cited on page 92.)

[GLC20] Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. *Advances in Neural Information Processing Systems*, 33:85–95, 2020. (Cited on page 4.)

[GRCvdM18] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. (Cited on page 18.)

[GSS14a] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. dec 2014. (Cited on pages 2, 3, 5, 41, 52, 62, 68, 78, and 111.)

[GSS14b] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. (Cited on pages 5 and 95.)

[HD19] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. (Cited on page 111.)

[HDY$^+$12] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and Brian Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *Ieee Signal Processing Magazine*, 2012. (Cited on pages 62 and 78.)

[HK22] Yi Huang and Adams Wai-Kin Kong. Transferable adversarial attack based on integrated gradients. *arXiv preprint arXiv:2205.13152*, 2022. (Cited on page 4.)

[HMC+19] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019. (Cited on page 111.)

[HMP+21] Tianjin Huang, Vlado Menkovski, Yulong Pei, YuHao Wang, and Mykola Pechenizkiy. Direction-aggregated attack for transferable adversarial examples. *arXiv preprint arXiv:2104.09172*, 2021. (Cited on page 41.)

[HMP+22] Tianjin Huang, Vlado Menkovski, Yulong Pei, Yuhao Wang, and Mykola Pechenizkiy. Direction-aggregated attack for transferable adversarial examples. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 18(3):1–22, 2022. (Cited on page 4.)

[HMPP20] Tianjin Huang, Vlado Menkovski, Yulong Pei, and Mykola Pechenizkiy. Bridging the performance gap between fgsm and pgd adversarial training. *arXiv preprint arXiv:2011.05157*, 2020. (Cited on pages 6 and 44.)

[HMPP21] Tianjin Huang, Vlado Menkovski, Yulong Pei, and Mykola Pechenizkiy. calibrated adversarial training. In *Asian Conference on Machine Learning*, pages 626–641. PMLR, 2021. (Cited on page 5.)

[HRS16] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016. (Cited on page 62.)

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. (Cited on pages 14, 23, 41, 52, 62, 67, 78, and 84.)

[IEAL18] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information.

In *International Conference on Machine Learning*, pages 2137–2146. PMLR, 2018. (Cited on page 4.)

[IPG^{+}18] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. (Cited on page 66.)

[IWLC19] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7066–7074, 2019. (Cited on page 17.)

[JBZB19] Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. In *International Conference on Learning Representations*, 2019. (Cited on page 42.)

[JG18] Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 514–529, 2018. (Cited on pages 42 and 78.)

[JNM^{+}19] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019. (Cited on pages 6 and 64.)

[JWCF19] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6084–6092, 2019. (Cited on pages 23 and 27.)

[JZ19] Hongwei Jin and Xinhua Zhang. Latent adversarial training of graph convolution networks. In *ICML Workshop on Learning and Reasoning with Graph-Structured Representations*, 2019. (Cited on page 93.)

[KDS19] Sandesh Kamath, Amit Deshpande, and KV Subrahmanyam. Invariance vs robustness of neural networks. 2019. (Cited on page 111.)

[KGB16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. (Cited on pages 3, 19, and 20.)

[KGB17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017. (Cited on pages 4, 14, 16, 78, 86, and 87.)

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. (Cited on page 84.)

[KH12] Alex Krizhevsky and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 2012. (Cited on pages 14, 41, and 78.)

[KKG18] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018. (Cited on page 43.)

[KNH10] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 5, 2010. (Cited on pages 52 and 66.)

[KW16a] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. (Cited on page 92.)

[KW16b] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016. (Cited on pages 92, 94, 98, and 99.)

[LBB+98] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. (Cited on page 84.)

[LBZ+20] Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. Learning transferable adversarial examples via ghost networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11458–11465, 2020. (Cited on page 17.)

[LCLS17] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017. (Cited on pages 4, 17, 27, 78, 84, and 89.)

[LCO19] Alfred Laugros, Alice Caplier, and Matthieu Ospici. Are adversarial robustness and common perturbation robustness independant attributes? In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. (Cited on page 111.)

[LCWC19] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive gaussian noise. In *Neurips 2019*, 2019. (Cited on page 14.)

[LeC98] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. (Cited on page 52.)

[lGS15] J. Shlens l. Goodfellow and C. Szegedy. explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. (Cited on pages 14, 17, and 19.)

[LLD$^+$18] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2018. (Cited on pages 5, 18, 23, and 27.)

[LLL$^+$19] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–868. IEEE, 2019. (Cited on pages 23 and 27.)

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation ppt. In *CVPR 2015 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. (Cited on pages 14, 41, and 78.)

[LSH$^+$20] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2020. (Cited on pages 4, 17, 22, 23, 24, and 27.)

[MDFF16] Seyed Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 2574–2582. IEEE Computer Society, dec 2016. (Cited on pages 2, 3, 14, 17, 78, and 84.)

[MDFFF17] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. (Cited on pages 2 and 4.)

[MDFUF18] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. nov 2018. (Cited on pages 42, 74, 78, 83, 85, 86, and 87.)

[MDFUF19] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9078–9086, 2019. (Cited on page 5.)

[MMS+17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. (Cited on pages 3, 5, 62, 68, 92, 95, and 96.)

[MMS+18] 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*, 2018. (Cited on pages 2, 5, 14, 17, 42, 44, 52, 78, 80, 87, and 111.)

[MSMH+19] Naseer Muzammal, Khan Salman, Khan Muhammad Haris, Shahbaz Khan Fahad, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. In *Thirty-Third Conference on Neural information Processing Systems (NeurIPS 2019), Vancouver, Canada*, 2019. (Cited on page 17.)

[Nak19] Preetum Nakkiran. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*, 2019. (Cited on page 5.)

[Nes83] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983. (Cited on page 111.)

[NWC+11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. (Cited on page 66.)

[PARS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014. (Cited on page 92.)

[PHL+18] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. *arXiv preprint arXiv:1802.04407*, 2018. (Cited on page 99.)

[PHvIP20] Yulong Pei, Tianjin Huang, Werner van Ipenburg, and Mykola Pechenizkiy. Resgcn: Attention-based deep residual modeling for anomaly detection on attributed networks. *arXiv preprint arXiv:2009.14738*, 2020. (Cited on pages 92, 94, 101, and 102.)

[PLY+22] Tianyu Pang, Min Lin, Xiao Yang, Jun Zhu, and Shuicheng Yan. Robustness and accuracy could be reconcilable by (proper) definition. *arXiv preprint arXiv:2202.10103*, 2022. (Cited on page 5.)

[PMG+17] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM, 2017. (Cited on page 78.)

[PMW+16] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, pages 582–597.

Institute of Electrical and Electronics Engineers Inc., aug 2016. (Cited on pages 3 and 78.)

[QLZW19] Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. Review of artificial intelligence adversarial attack and defense technologies, 2019. (Cited on page 78.)

[QMG+19] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy, Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial Robustness through Local Linearization. jul 2019. (Cited on pages 4 and 5.)

[RBB17] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017. (Cited on page 84.)

[RDV18] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. (Cited on page 5.)

[RGC+21] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021. (Cited on pages 62 and 63.)

[RWK20] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020. (Cited on pages 6, 62, 63, and 67.)

[RXY+20] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*, 2020. (Cited on page 5.)

[SAP+20] 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. *arXiv preprint arXiv:2003.05669*, 2020. (Cited on page 102.)

[SCW20] Chawin Sitawarin, Supriyo Chakraborty, and David Wagner. Improving adversarial robustness through progressive hardening. *arXiv preprint arXiv:2003.09347*, 2020. (Cited on pages 44 and 111.)

[SDB19] Yash Sharma, Gavin Weiguang Ding, and Marcus Brubaker. On the effectiveness of low frequency perturbations. *arXiv preprint arXiv:1903.00073*, 2019. (Cited on page 42.)

[SEE⁺18] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX workshop on offensive technologies (WOOT 18)*, 2018. (Cited on page 4.)

[SFK20] Han Shi, Haozheng Fan, and James T Kwok. Effective decoding in graph auto-encoder using triadic closure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 906–913, 2020. (Cited on pages 92 and 94.)

[SHS19] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019. (Cited on page 93.)

[SHS21] David Stutz, Matthias Hein, and Bernt Schiele. Relating adversarially robust generalization to flat minima. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7807–7817, 2021. (Cited on pages 6, 62, 63, 64, 69, and 74.)

[SHV20] Guillaume Salha, Romain Hennequin, and Michalis Vazirgiannis. Simple and effective graph autoencoders with one-hop linear models. *arXiv preprint arXiv:2001.07614*, 2020. (Cited on pages 92 and 94.)

[SIE⁺20] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *arXiv preprint arXiv:2007.08489*, 2020. (Cited on pages 92 and 93.)

[SIVA17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. (Cited on page 23.)

[SKB$^+$18] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018. (Cited on page 92.)

[SKC18] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. (Cited on page 5.)

[SKN$^+$17] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples. oct 2017. (Cited on page 78.)

[SLGZ19] Ke Sun, Zhouchen Lin, Hantao Guo, and Zhanxing Zhu. Virtual adversarial training on graph convolutional networks in node classification. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 431–443. Springer, 2019. (Cited on page 93.)

[SLH$^+$19] Guillaume Salha, Stratis Limnios, Romain Hennequin, Viet-Anh Tran, and Michalis Vazirgiannis. Gravity-inspired graph autoencoders for directed link prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 589–598, 2019. (Cited on page 94.)

[SLL21] Ke Sun, Mingjie Li, and Zhouchen Lin. Pareto adversarial robustness: Balancing spatial robustness and sensitivity-based robustness. *arXiv preprint arXiv:2111.01996*, 2021. (Cited on page 111.)

[SMH$^+$21] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? *arXiv preprint arXiv:2104.09425*, 2021. (Cited on page 62.)

[SNG$^+$19] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and

Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019. (Cited on page 6.)

[SSFJ21] Vasu Singla, Sahil Singla, Soheil Feizi, and David Jacobs. Low curvature activations reduce overfitting in adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16423–16433, 2021. (Cited on pages 6 and 63.)

[SST$^+$18] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pages 5014–5026, 2018. (Cited on pages 5, 45, and 62.)

[SVI$^+$16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. (Cited on page 23.)

[SVS19] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019. (Cited on page 4.)

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. (Cited on page 67.)

[SZC$^+$18] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018. (Cited on page 5.)

[SZS$^+$13] 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*, 2013. (Cited on pages 1, 2, 3, 4, 5, 14, 16, 20, 41, 62, and 78.)

[TB19] Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. *Advances in Neural Information Processing Systems*, 32, 2019. (Cited on page 111.)

[TBC+20] Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jörn-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. *arXiv preprint arXiv:2002.04599*, 2020. (Cited on page 42.)

[TGC+14] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014. (Cited on pages 92 and 94.)

[TKP+18] 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*, 2018. (Cited on pages 18 and 23.)

[TKRB19] Timothy Tadros, Giri Krishnan, Ramyaa Ramyaa, and Maxim Bazhenov. Biologically inspired sleep algorithm for increased generalization and adversarial robustness in deep neural networks. In *International Conference on Learning Representations*, 2019. (Cited on page 42.)

[TQW+15] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015. (Cited on page 92.)

[TSE+18a] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness May Be at Odds with Accuracy. may 2018. (Cited on pages 5 and 87.)

[TSE+18b] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. (Cited on page 93.)

[UKE+20] Francisco Utrera, Evan Kravitz, N Benjamin Erichson, Rajiv Khanna, and Michael W Mahoney. Adversarially-trained deep nets transfer better. *arXiv preprint arXiv:2007.05869*, 2020. (Cited on pages 92 and 93.)

[UOKO18] Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018. (Cited on page 68.)

[WCG$^+$20] Haotao Wang, Tianlong Chen, Shupeng Gui, TingKuei Hu, Ji Liu, and Zhangyang Wang. Once-for-all adversarial training: In-situ tradeoff between robustness and accuracy for free. *Advances in Neural Information Processing Systems*, 33:7449–7461, 2020. (Cited on page 5.)

[WCZ16] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234, 2016. (Cited on page 92.)

[WH21] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1924–1933, 2021. (Cited on page 4.)

[WK18] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018. (Cited on page 42.)

[WLH19] Xiaoyun Wang, Xuanqing Liu, and Cho-Jui Hsieh. Graphdefense: Towards robust graph convolutional networks. *arXiv preprint arXiv:1911.04429*, 2019. (Cited on page 93.)

[WMB$^+$19] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *International Conference on Machine Learning*, pages 6586–6595, 2019. (Cited on pages 44 and 78.)

[WMG17] Marty J Wolf, Keith W Miller, and Frances S Grodzinsky. Why we should have seen that coming: comments on microsoft's tay "experiment," and wider implications. *The ORBIT Journal*, 1(2):1–12, 2017. (Cited on page 1.)

[WRK20] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020. (Cited on page 6.)

[WWX20a] Dongxian Wu, Yisen Wang, and Shu-tao Xia. Revisiting loss landscape for adversarial robustness. *arXiv preprint arXiv:2004.05884*, 2020. (Cited on page 63.)

[WWX+20b] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. *arXiv preprint arXiv:2002.05990*, 2020. (Cited on page 4.)

[WXW20] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33, 2020. (Cited on pages 6, 44, 52, 62, 63, 64, 67, 69, 73, and 74.)

[WZ20] Lei Wu and Zhanxing Zhu. Towards understanding and improving the transferability of adversarial examples in deep neural networks. In *Asian Conference on Machine Learning*, pages 837–850. PMLR, 2020. (Cited on pages 4 and 17.)

[WZY+20] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020. (Cited on pages 43, 51, 52, 59, 67, and 95.)

[XCL+19] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: An optimization perspective. *arXiv preprint arXiv:1906.04214*, 2019. (Cited on page 93.)

[XPDY14] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014. (Cited on page 99.)

[XWZ+18] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. (Cited on pages 18, 23, and 27.)

[XZL+18] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In

International Conference on Learning Representations, 2018. (Cited on page 4.)

[XZZ$^+$19] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. (Cited on pages 2, 4, 17, 20, 23, 24, 41, and 111.)

[YHG$^+$21] Chaojian Yu, Bo Han, Mingming Gong, Li Shen, Shiming Ge, Bo Du, and Tongliang Liu. Robust weight perturbation for adversarial training. 2021. (Cited on pages 6, 62, and 64.)

[YZJ$^+$19] Donghan Yu, Ruohong Zhang, Zhengbao Jiang, Yuexin Wu, and Yiming Yang. Graph-revised convolutional network. *arXiv preprint arXiv:1911.07123*, 2019. (Cited on page 92.)

[ZHC$^+$18] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467, 2018. (Cited on page 17.)

[ZK16] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. (Cited on pages 52, 67, and 84.)

[ZL19] Jiliang Zhang and Chen Li. Adversarial examples: Opportunities and challenges. *IEEE transactions on neural networks and learning systems*, 31(7):2578–2593, 2019. (Cited on page 3.)

[ZLR$^+$17] Chiyuan Zhang, Qianli Liao, Alexander Rakhlin, Karthik Sridharan, Brando Miranda, Noah Golowich, and Tomaso Poggio. Musings on deep learning: Properties of sgd. Technical report, Center for Brains, Minds and Machines (CBMM), 2017. (Cited on page 62.)

[ZLZ18] Yi Zhou, Yingbin Liang, and Huishuai Zhang. Generalization error bounds with probabilistic guarantee for sgd in nonconvex optimization. *arXiv preprint arXiv:1802.06903*, 2018. (Cited on page 62.)

[ZXH$^+$20a] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International conference on machine learning*, pages 11278–11287. PMLR, 2020. (Cited on pages 5 and 111.)

[ZXH$^+$20b] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11278–11287. PMLR, 13–18 Jul 2020. (Cited on pages 44 and 52.)

[ZYJ$^+$19a] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019. (Cited on pages 5, 67, 95, and 102.)

[ZYJ$^+$19b] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 09–15 Jun 2019. (Cited on pages 43, 45, 49, 51, 52, 58, and 59.)

[ZZG$^+$20] Haizhong Zheng, Ziqi Zhang, Juncheng Gu, Honglak Lee, and Atul Prakash. Efficient adversarial training with transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1181–1190, 2020. (Cited on page 6.)

[ZZL$^+$19] Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. *Advances in Neural Information Processing Systems*, 32, 2019. (Cited on page 6.)

[ZZN$^+$20] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instancereweighted adversarial training. *arXiv preprint arXiv:2010.01736*, 2020. (Cited on page 5.)

Curriculum Vitae

Tianjin Huang was born on February 11, 1990 in Tongling, China. He completed a bachelor's degree in Cartography and Geographic Information System at Northwest University in 2014 and went on to earn an MSc in the Institute of Remote Sensing and Digital Earth at the University of Chinese Academy of Sciences in 2017. In 2018, he moved to the Netherlands and began a Ph.D. study in the data mining group at Eindhoven University of Technology, where he published 17 research papers, with 8 of them being his first-author publications. He is an active member of the scientific community, serving as a reviewer for conferences and journals such as ICML, NeurIPS, ECML, and IEEE TII. He received **the best paper award** as the first author of "You Can Have Better Graph Neural Networks by Not Training Weights at All: Finding Untrained GNNs Tickets" at the Learning on Graphs (LoG) 2022 Conference.

Acknowledgments

I have been very fortunate to fill the past four years with extremely valuable experiences that helped me to grow personally and professionally. During the course of this Ph.D., I have interacted with and learned from many excellent people. I would like to take the chance to express my sincere gratitude.

First and foremost, I want to express heartfelt gratitude to my great promoter, Prof. Mykola Pechenizkiy. Thank you for the countless help you provided in both my research and personal life during my stay in Eindhoven. Thank you for your never-ending support and encouragement for my Ph.D. studies. Thank you for affording me the opportunity to pursue my own research interests and ideas while offering insightful feedback. Thank you for providing a financial guarantee to the leasing company in the year 2019, allowing me to have a place to reside. Thank you for giving me a contract after my scholarship ended, enabling me to continue my Ph.D. studies. Thank you for your guidance in balancing work and life. There is still much help that goes unmentioned, but it remains etched in my heart. Your immense knowledge and plentiful experience have encouraged me all the time in my Ph.D. journey. Without your support and encouragement, I can not complete my Ph.D. Thank you from the bottom of my heart, Professor Pechenizkiy.

I would like to express my gratitude to my co-promoters, dr. Vlado Menkovski and dr. Yulong Pei. Thank you for the constant support and constructive discussion in developing research ideas and directions. I would like to thank Vlado for his wise, patient, and comprehensive suggestions in my Ph.D. research. I would also like to appreciate Vlado for explaining the dutch culture and offering wonderful travel destinations to me, facilitating my quick adaption to dutch life. I want to thank Yulong for his unwavering support in my personal life and Ph.D. studies. Particularly, I would like to express sincere gratitude to him for his great help during my Covid positive period.

I would like to take the chance to thank the committee members of this Ph.D. defense, for reviewing this dissertation and for your insightful comments: prof.dr. F. Roli from the University of Cagliari, prof.dr. X. Huang from the University of Liverpool, prof.dr.ir. W.P.A.J. Michiels and Dr.habil. C. De Campos from Technische Universiteit Eindhoven, and M. Fang from Technische Universiteit Eindhoven. Your insightful and detailed comments have greatly enhanced the completeness of my text and helped me to clarify the research questions better.

My sincere appreciation goes to José, Ine, and Mariela for their unwavering help and supportive endeavors. Your diligent work and kind hearts have made me feel like I'm at home, even while studying abroad.

I am grateful to Zhaohuan, Jianpeng, Cong, Shiwei, and Guangming for welcoming me to the group when I first came to TU/e. I also appreciate other friends and colleagues including Meng, Lu, Kaijie, Qiao, Boqian, Jiaxu, Zhao, Yucheng, Wenhan, Ziwei, Yudi, Mengting, Yuhao, Linhao, Fangqin and Yang for regular conversations, movie nights, and hotpot parties.

I would like to thank dr. Wouter Duivesteijn for the invitation to volunteer in IDA 2018, which was my first international conference experience. Thank you for inviting me to your romantic wedding at Strandclub Naturel, birthday party, and poker nights. I would also thank other colleagues including Tristan, Onur, Ceren, Bram, Stiven, Simon, Ricky, Tita, Pieter, Loek, Bilge, Marign, Pratik, and all the ones not explicitly mentioned here. All the conversations and discussions with you make my Ph.D. life more memorable and fun.

I would like to extend my sincere gratitude to my great collaborators dr. Shiwei Liu, dr. Meng Fang, dr. Sibylle, dr. Li Shen, dr. Decebal C. Mocanu, dr. Atlas Wang, Tianlong Chen, Lu Yin and Yutian Liu. I feel honored to have had the opportunity to work with each and every one of you, and I am deeply grateful for your collaboration and support. Special thanks to Shiwei, Tianlong, Meng, Decebal, and Atlas for their exceptional contributions to our collaborated paper that enabled me to receive the best paper award at the LoG conference 2022.

Last but not least, I want to express my deepest gratitude to my beloved brother and parents. Your unwavering support and understanding have been a constant throughout my life. Your love and compassion have always been a source of strength for me.

Tianjin Huang
Eindhoven, February, 2023

SIKS Dissertations

2011

01 Botond Cseke (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models

02 Nick Tinnemeier (UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

03 Jan Martijn van der Werf (TUE), Compositional Design and Verification of Component-Based Information Systems

04 Hado van Hasselt (UU), Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference

05 Bas van der Raadt (VU), Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

06 Yiwen Wang (TUE), Semantically-Enhanced Recommendations in Cultural Heritage

07 Yujia Cao (UT), Multimodal Information Presentation for High Load Human Computer Interaction

08 Nieske Vergunst (UU), BDI-based Generation of Robust Task-Oriented Dialogues

09 Tim de Jong (OU), Contextualised Mobile Media for Learning

10 Bart Bogaert (UvT), Cloud Content Contention

11 Dhaval Vyas (UT), Designing for Awareness: An Experience-focused HCI Perspective

12 Carmen Bratosin (TUE), Grid Architecture for Distributed Process Mining

13 Xiaoyu Mao (UvT), Airport under Control. Multiagent Scheduling for Airport Ground Handling

14 Milan Lovric (EUR), Behavioral Finance and Agent-Based Artificial Markets

15 Marijn Koolen (UvA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval

16 Maarten Schadd (UM), Selective Search in Games of Different Complexity

17 Jiyin He (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness

18 Mark Ponsen (UM), Strategic Decision-Making in complex games

19 Ellen Rusman (OU), The Mind’s Eye on Personal Profiles

20 Qing Gu (VU), Guiding service-oriented software engineering - A view-based approach

21 Linda Terlouw (TUD), Modularization and Specification of Service-Oriented Systems

22 Junte Zhang (UVA), System Evaluation of Archival Description and Access

23 Wouter Weerkamp (UVA), Finding People and their Utterances in Social Media

24 Herwin van Welbergen (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

25 Syed Waqar ul Qounain Jaffry (VU), Analysis and Validation of Models for Trust Dynamics

26 Matthijs Aart Pontier (VU), Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

27 Aniel Bhulai (VU), Dynamic website optimization through autonomous management of design patterns

28 Rianne Kaptein (UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure

29 Faisal Kamiran (TUE), Discrimination-aware Classification

30 Egon van den Broek (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions

31 Ludo Waltman (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality

32 Nees-Jan van Eck (EUR), Methodological Advances in Bibliometric Mapping of Science

33 Tom van der Weide (UU), Arguing to Motivate Decisions

34 Paolo Turrini (UU), Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations

35 Maaike Harbers (UU), Explaining Agent Behavior in Virtual Training

36 Erik van der Spek (UU), Experiments in serious game design: a cognitive approach

37 Adriana Burlutiu (RUN), Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference

38 Nyree Lemmens (UM), Bee-inspired Distributed Optimization

39 Joost Westra (UU), Organizing Adaptation using Agents in Serious Games

40 Viktor Clerc (VU), Architectural Knowledge Management in Global Software Development

41 Luan Ibraimi (UT), Cryptographically Enforced Distributed Data Access Control

42 Michal Sindlar (UU), Explaining Behavior through Mental State Attribution

43 Henk van der Schuur (UU), Process Improvement through Software Operation Knowledge

44 Boris Reuderink (UT), Robust Brain-Computer Interfaces

45 Herman Stehouwer (UvT), Statistical Language Models for Alternative Sequence Selection

46 Beibei Hu (TUD), Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work

47 Azizi Bin Ab Aziz (VU), Exploring Computational Models for Intelligent Support of Persons with Depression

48 Mark Ter Maat (UT), Response Selection and Turn-taking for a Sensitive Artificial Listening Agent

49 Andreea Niculescu (UT), Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

---

2012 01 Terry Kakeeto (UvT), Relationship Marketing for SMEs in Uganda

02 Muhammad Umair (VU), Adaptivity, emotion, and Rationality in Human and Ambient Agent Models

03 Adam Vanya (VU), Supporting Architecture Evolution by Mining Software Repositories

04 Jurriaan Souer (UU), Development of Content Management System-based Web Applications

05 Marijn Plomp (UU), Maturing Interorganisational Information Systems

06 Wolfgang Reinhardt (OU), Awareness Support for Knowledge Workers in Research Networks

07 Rianne van Lambalgen (VU), When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions

08 Gerben de Vries (UVA), Kernel Methods for Vessel Trajectories

09 Ricardo Neisse (UT), Trust and Privacy Management Support for Context-Aware Service Platforms
10 David Smits (TUE), Towards a Generic Distributed Adaptive Hypermedia Environment
11 J.C.B. Rantham Prabhakara (TUE), Process Mining in the Large: Preprocessing, Discovery, and Diagnostics
12 Kees van der Sluijs (TUE), Model Driven Design and Data Integration in Semantic Web Information Systems
13 Suleman Shahid (UvT), Fun and Face: Exploring non-verbal expressions of emotion during playful interactions
14 Evgeny Knutov (TUE), Generic Adaptation Framework for Unifying Adaptive Web-based Systems
15 Natalie van der Wal (VU), Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.
16 Fiemke Both (VU), Helping people by understanding them - Ambient Agents supporting task execution and depression treatment
17 Amal Elgammal (UvT), Towards a Comprehensive Framework for Business Process Compliance
18 Eltjo Poort (VU), Improving Solution Architecting Practices
19 Helen Schonenberg (TUE), What's Next? Operational Support for Business Process Execution
20 Ali Bahramisharif (RUN), Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing
21 Roberto Cornacchia (TUD), Querying Sparse Matrices for Information Retrieval
22 Thijs Vis (UvT), Intelligence, politie en veiligheidsdienst: verenigbare grootheden?
23 Christian Muehl (UT), Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
24 Laurens van der Werff (UT), Evaluation of Noisy Transcripts for Spoken Document Retrieval
25 Silja Eckartz (UT), Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application
26 Emile de Maat (UVA), Making Sense of Legal Text
27 Hayrettin Gurkok (UT), Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games

28 Nancy Pascall (UvT), Engendering Technology Empowering Women

29 Almer Tigelaar (UT), Peer-to-Peer Information Retrieval

30 Alina Pommeranz (TUD), Designing Human-Centered Systems for Reflective Decision Making

31 Emily Bagarukayo (RUN), A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure

32 Wietske Visser (TUD), Qualitative multi-criteria preference representation and reasoning

33 Rory Sie (OUN), Coalitions in Cooperation Networks (COCOON)

34 Pavol Jancura (RUN), Evolutionary analysis in PPI networks and applications

35 Evert Haasdijk (VU), Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics

36 Denis Ssebugwawo (RUN), Analysis and Evaluation of Collaborative Modeling Processes

37 Agnes Nakakawa (RUN), A Collaboration Process for Enterprise Architecture Creation

38 Selmar Smit (VU), Parameter Tuning and Scientific Testing in Evolutionary Algorithms

39 Hassan Fatemi (UT), Risk-aware design of value and coordination networks

40 Agus Gunawan (UvT), Information Access for SMEs in Indonesia

41 Sebastian Kelle (OU), Game Design Patterns for Learning

42 Dominique Verpoorten (OU), Reflection Amplifiers in self-regulated Learning

43 Withdrawn

44 Anna Tordai (VU), On Combining Alignment Techniques

45 Benedikt Kratz (UvT), A Model and Language for Business-aware Transactions

46 Simon Carter (UVA), Exploration and Exploitation of Multilingual Data for Statistical Machine Translation

47 Manos Tsagkias (UVA), Mining Social Media: Tracking Content and Predicting Behavior

48 Jorn Bakker (TUE), Handling Abrupt Changes in Evolving Time-series Data

49 Michael Kaisers (UM), Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions

50 Steven van Kervel (TUD), Ontologogy driven Enterprise Information Systems Engineering

51 Jeroen de Jong (TUD), Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

---

2013

01 Viorel Milea (EUR), News Analytics for Financial Decision Support

02 Erietta Liarou (CWI), MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing

03 Szymon Klarman (VU), Reasoning with Contexts in Description Logics

04 Chetan Yadati (TUD), Coordinating autonomous planning and scheduling

05 Dulce Pumareja (UT), Groupware Requirements Evolutions Patterns

06 Romulo Goncalves (CWI), The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience

07 Giel van Lankveld (UvT), Quantifying Individual Player Differences

08 Robbert-Jan Merk (VU), Making enemies: cognitive modeling for opponent agents in fighter pilot simulators

09 Fabio Gori (RUN), Metagenomic Data Analysis: Computational Methods and Applications

10 Jeewanie Jayasinghe Arachchige (UvT), A Unified Modeling Framework for Service Design.

11 Evangelos Pournaras (TUD), Multi-level Reconfigurable Self-organization in Overlay Services

12 Marian Razavian (VU), Knowledge-driven Migration to Services

13 Mohammad Safiri (UT), Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly

14 Jafar Tanha (UVA), Ensemble Approaches to Semi-Supervised Learning Learning

15 Daniel Hennes (UM), Multiagent Learning - Dynamic Games and Applications

16 Eric Kok (UU), Exploring the practical benefits of argumentation in multi-agent deliberation

17 Koen Kok (VU), The PowerMatcher: Smart Coordination for the Smart Electricity Grid

18 Jeroen Janssens (UvT), Outlier Selection and One-Class Classification

19 Renze Steenhuizen (TUD), Coordinated Multi-Agent Planning and Scheduling

20 Katja Hofmann (UvA), Fast and Reliable Online Learning to Rank for Information Retrieval

21 Sander Wubben (UvT), Text-to-text generation by monolingual machine translation

22 Tom Claassen (RUN), Causal Discovery and Logic

23 Patricio de Alencar Silva (UvT), Value Activity Monitoring

24 Haitham Bou Ammar (UM), Automated Transfer in Reinforcement Learning

25 Agnieszka Anna Latoszek-Berendsen (UM), Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System

26 Alireza Zarghami (UT), Architectural Support for Dynamic Homecare Service Provisioning

27 Mohammad Huq (UT), Inference-based Framework Managing Data Provenance

28 Frans van der Sluis (UT), When Complexity becomes Interesting: An Inquiry into the Information eXperience

29 Iwan de Kok (UT), Listening Heads

30 Joyce Nakatumba (TUE), Resource-Aware Business Process Management: Analysis and Support

31 Dinh Khoa Nguyen (UvT), Blueprint Model and Language for Engineering Cloud Applications

32 Kamakshi Rajagopal (OUN), Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development

33 Qi Gao (TUD), User Modeling and Personalization in the Microblogging Sphere

34 Kien Tjin-Kam-Jet (UT), Distributed Deep Web Search

35 Abdallah El Ali (UvA), Minimal Mobile Human Computer Interaction

36 Than Lam Hoang (TUe), Pattern Mining in Data Streams

37 Dirk Börner (OUN), Ambient Learning Displays

38 Eelco den Heijer (VU), Autonomous Evolutionary Art

39 Joop de Jong (TUD), A Method for Enterprise Ontology based Design of Enterprise Information Systems

40 Pim Nijssen (UM), Monte-Carlo Tree Search for Multi-Player Games

41 Jochem Liem (UVA), Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning

42 Léon Planken (TUD), Algorithms for Simple Temporal Reasoning

43 Marc Bron (UVA), Exploration and Contextualization through Interaction and Concepts

---

2014 01 Nicola Barile (UU), Studies in Learning Monotone Models from Data

02 Fiona Tuliyano (RUN), Combining System Dynamics with a Domain Modeling Method

03 Sergio Raul Duarte Torres (UT), Information Retrieval for Children: Search Behavior and Solutions

04 Hanna Jochmann-Mannak (UT), Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation

05 Jurriaan van Reijsen (UU), Knowledge Perspectives on Advancing Dynamic Capability

06 Damian Tamburri (VU), Supporting Networked Software Development

07 Arya Adriansyah (TUE), Aligning Observed and Modeled Behavior

08 Samur Araujo (TUD), Data Integration over Distributed and Heterogeneous Data Endpoints

09 Philip Jackson (UvT), Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language

10 Ivan Salvador Razo Zapata (VU), Service Value Networks

11 Janneke van der Zwaan (TUD), An Empathic Virtual Buddy for Social Support

12 Willem van Willigen (VU), Look Ma, No Hands: Aspects of Autonomous Vehicle Control

13 Arlette van Wissen (VU), Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains

14 Yangyang Shi (TUD), Language Models With Meta-information

15 Natalya Mogles (VU), Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare

16 Krystyna Milian (VU), Supporting trial recruitment and design by automatically interpreting eligibility criteria

17 Kathrin Dentler (VU), Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability

18 Mattijs Ghijsen (UVA), Methods and Models for the Design and Study of Dynamic Agent Organizations

19 Vinicius Ramos (TUE), Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support

20 Mena Habib (UT), Named Entity Extraction and Disambiguation for Informal Text: The Missing Link

21 Kassidy Clark (TUD), Negotiation and Monitoring in Open Environments

22 Marieke Peeters (UU), Personalized Educational Games - Developing agent-supported scenario-based training

23 Eleftherios Sidirourgos (UvA/CWI), Space Efficient Indexes for the Big Data Era

24 Davide Ceolin (VU), Trusting Semi-structured Web Data

25 Martijn Lappenschaar (RUN), New network models for the analysis of disease interaction

26 Tim Baarslag (TUD), What to Bid and When to Stop

27 Rui Jorge Almeida (EUR), Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty

28 Anna Chmielowiec (VU), Decentralized k-Clique Matching

29 Jaap Kabbedijk (UU), Variability in Multi-Tenant Enterprise Software

30 Peter de Cock (UvT), Anticipating Criminal Behaviour

31 Leo van Moergestel (UU), Agent Technology in Agile Multiparallel Manufacturing and Product Support

32 Naser Ayat (UvA), On Entity Resolution in Probabilistic Data

33 Tesfa Tegegne (RUN), Service Discovery in eHealth

34 Christina Manteli (VU), The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.

35 Joost van Ooijen (UU), Cognitive Agents in Virtual Worlds: A Middleware Design Approach

36 Joos Buijs (TUE), Flexible Evolutionary Algorithms for Mining Structured Process Models

37 Maral Dadvar (UT), Experts and Machines United Against Cyberbullying

38 Danny Plass-Oude Bos (UT), Making brain-computer interfaces better: improving usability through post-processing.

39 Jasmina Maric (UvT), Web Communities, Immigration, and Social Capital

40 Walter Omona (RUN), A Framework for Knowledge Management Using ICT in Higher Education

41 Frederic Hogenboom (EUR), Automated Detection of Financial Events in News Text

42 Carsten Eijckhof (CWI/TUD), Contextual Multidimensional Relevance Models

43 Kevin Vlaanderen (UU), Supporting Process Improvement using Method Increments

44 Paulien Meesters (UvT), Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.

45 Birgit Schmitz (OUN), Mobile Games for Learning: A Pattern-Based Approach

46 Ke Tao (TUD), Social Web Data Analytics: Relevance, Redundancy, Diversity

47 Shangsong Liang (UVA), Fusion and Diversification in Information Retrieval

---

2015 01 Niels Netten (UvA), Machine Learning for Relevance of Information in Crisis Response

02 Faiza Bukhsh (UvT), Smart auditing: Innovative Compliance Checking in Customs Controls

03 Twan van Laarhoven (RUN), Machine learning for network data

04 Howard Spoelstra (OUN), Collaborations in Open Learning Environments

05 Christoph Bösch (UT), Cryptographically Enforced Search Pattern Hiding

06 Farideh Heidari (TUD), Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes

07 Maria-Hendrike Peetz (UvA), Time-Aware Online Reputation Analysis

08 Jie Jiang (TUD), Organizational Compliance: An agent-based model for designing and evaluating organizational interactions

09 Randy Klaassen (UT), HCI Perspectives on Behavior Change Support Systems

10 Henry Hermans (OUN), OpenU: design of an integrated system to support lifelong learning
11 Yongming Luo (TUE), Designing algorithms for big graph datasets: A study of computing bisimulation and joins
12 Julie M. Birkholz (VU), Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks
13 Giuseppe Procaccianti (VU), Energy-Efficient Software
14 Bart van Straalen (UT), A cognitive approach to modeling bad news conversations
15 Klaas Andries de Graaf (VU), Ontology-based Software Architecture Documentation
16 Changyun Wei (UT), Cognitive Coordination for Cooperative Multi-Robot Teamwork
17 André van Cleeff (UT), Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs
18 Holger Pirk (CWI), Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories
19 Bernardo Tabuenca (OUN), Ubiquitous Technology for Lifelong Learners
20 Lois Vanhée (UU), Using Culture and Values to Support Flexible Coordination
21 Sibren Fetter (OUN), Using Peer-Support to Expand and Stabilize Online Learning
22 Zhemin Zhu (UT), Co-occurrence Rate Networks
23 Luit Gazendam (VU), Cataloguer Support in Cultural Heritage
24 Richard Berendsen (UVA), Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation
25 Steven Woudenberg (UU), Bayesian Tools for Early Disease Detection
26 Alexander Hogenboom (EUR), Sentiment Analysis of Text Guided by Semantics and Structure
27 Sándor Héman (CWI), Updating compressed colomn stores
28 Janet Bagorogoza (TiU), Knowledge Management and High Performance; The Uganda Financial Institutions Model for HPO
29 Hendrik Baier (UM), Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains
30 Kiavash Bahreini (OU), Real-time Multimodal Emotion Recognition in E-Learning
31 Yakup Koç (TUD), On the robustness of Power Grids

32 Jerome Gard (UL), Corporate Venture Management in SMEs

33 Frederik Schadd (TUD), Ontology Mapping with Auxiliary Resources

34 Victor de Graaf (UT), Gesocial Recommender Systems

35 Jungxao Xu (TUD), Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

---

2016

01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines

02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow

03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support

04 Laurens Rietveld (VU), Publishing and Consuming Linked Data

05 Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers

06 Michel Wilson (TUD), Robust scheduling in an uncertain environment

07 Jeroen de Man (VU), Measuring and modeling negative emotions for virtual training

08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data

09 Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts

10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms

11 Anne Schuth (UVA), Search Engines that Learn from Their Users

12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems

13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach

14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization

15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments

16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward

17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms

18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web
19 Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data
20 Daan Odijk (UVA), Context & Semantics in News & Web Search
21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems
23 Fei Cai (UVA), Query Auto Completion in Information Retrieval
24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach
25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media
28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control
29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30 Ruud Mattheij (UvT), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web content monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33 Peter Bloem (UVA), Single Sample Statistics, exercises in learning from just one example
34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment
35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry

	38	Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
	39	Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
	40	Christian Detweiler (TUD), Accounting for Values in Design
	41	Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance
	42	Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
	43	Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
	44	Thibault Sellam (UVA), Automatic Assistants for Database Exploration
	45	Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
	46	Jorge Gallego Perez (UT), Robots to Make you Happy
	47	Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
	48	Tanja Buttler (TUD), Collecting Lessons Learned
	49	Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
	50	Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains
2017	01	Jan-Jaap Oerlemans (UL), Investigating Cybercrime
	02	Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
	03	Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
	04	Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
	05	Mahdieh Shadi (UVA), Collaboration Behavior
	06	Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
	07	Roel Bertens (UU), Insight in Information: from Abstract to Anomaly

08 Rob Konijn (VU) , Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery

09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text

10 Robby van Delden (UT), (Steering) Interactive Play Behavior

11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment

12 Sander Leemans (TUE), Robust Process Mining with Guarantees

13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology

14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior

15 Peter Berck (RUN), Memory-Based Text Correction

16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines

17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution

18 Ridho Reinanda (UVA), Entity Associations for Search

19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility

21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)

22 Sara Magliacane (VU), Logics for causal inference under uncertainty

23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces

24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning

25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search

26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch

27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People’s Preferences, Perceptions and Behaviors

28 John Klein (VU), Architecture Practices for Complex Contexts

29 Adel Alhuraibi (UvT), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"

30 Wilma Latuny (UvT), The Power of Facial Expressions

31 Ben Ruijl (UL), Advances in computational methods for QFT calculations

32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives

33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity

34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics

35 Martine de Vos (VU), Interpreting natural science spreadsheets

36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging

37 Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy

38 Alex Kayal (TUD), Normative Social Applications

39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR

40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems

41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle

42 Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets

43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval

44 Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering

45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement

46 Jan Schneider (OU), Sensor-based Learning Support

47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration

48 Angel Suarez (OU), Collaborative inquiry-based learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations

02 Felix Mannhardt (TUE), Multi-perspective Process Mining

03 Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction

04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks

05 Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process

06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems

07 Jieting Luo (UU), A formal account of opportunism in multi-agent systems

08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems

09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations

10 Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology

11 Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks

12 Xixi Lu (TUE), Using behavioral context in process mining

13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future

14 Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters

15 Naser Davarzani (UM), Biomarker discovery in heart failure

16 Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children

17 Jianpeng Zhang (TUE), On Graph Sample Clustering

18 Henriette Nakad (UL), De Notaris en Private Rechtspraak

19 Minh Duc Pham (VUA), Emergent relational schemas for RDF

20 Manxia Liu (RUN), Time and Bayesian Networks

21 Aad Slootmaker (OUN), EMERGO: a generic platform for authoring and playing scenario-based serious games

22 Eric Fernandes de Mello Araujo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks

23 Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis

24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots

25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections

26 Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology

27 Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis

28 Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel

29 Yu Gu (UVT), Emotion Recognition from Mandarin Speech

30 Wouter Beek, The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

---

2019 01 Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification

02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty

03 Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data Sources

04 Ridho Rahmadi (RUN), Finding stable causal structures from clinical data

05 Sebastiaan van Zelst (TUE), Process Mining with Streaming Data

06 Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets

07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms

08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes

09 Fahimeh Alizadeh Moghaddam (UVA), Self-adaptation for energy efficiency in software systems

10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction

11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs

12 Jacqueline Heinerman (VU), Better Together

13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation

14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses

15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments

16 Guangming Li (TUE), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models

17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts

18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication

19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents

20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence

21 Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection

22 Martin van den Berg (VU), Improving IT Decisions with Enterprise Architecture

23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification

24 Anca Dumitrache (VU), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing

25 Emiel van Miltenburg (VU), Pragmatic factors in (automatic) image description

26 Prince Singh (UT), An Integration Platform for Synchromodal Transport

27 Alessandra Antonaci (OUN), The Gamification Design Process applied to (Massive) Open Online Courses

28 Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations

29 Daniel Formolo (VU), Using virtual agents for simulation and training of social skills in safety-critical circumstances

30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems

31 Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics

32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games

33 Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks

34 Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES

35 Lisa Facey-Shaw (OUN), Gamification with digital badges in learning programming

36 Kevin Ackermans (OUN), Designing Video-Enhanced Rubrics to Master Complex Skills

37 Jian Fang (TUD), Database Acceleration on FPGAs

38 Akos Kadar (OUN), Learning visually grounded and multilingual representations

---

2020 01 Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour

02 Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models

03 Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding

04 Maarten van Gompel (RUN), Context as Linguistic Bridges

05 Yulong Pei (TUE), On local and global structure mining

06 Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support

07 Wim van der Vegt (OUN), Towards a software architecture for reusable game components

08 Ali Mirsoleimani (UL),Structured Parallel Programming for Monte Carlo Tree Search

09 Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research

10 Alifah Syamsiyah (TUE), In-database Preprocessing for Process Mining

11 Sepideh Mesbah (TUD), Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models

12 Ward van Breda (VU), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment

13 Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming

14 Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases

15 Konstantinos Georgiadis (OUN), Smart CAT: Machine Learning for Configurable Assessments in Serious Games

16 Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling

17 Daniele Di Mitri (OUN), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences

	18	Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems
	19	Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems
	20	Albert Hankel (VU), Embedding Green ICT Maturity in Organisations
	21	Karine da Silva Miras de Araujo (VU), Where is the robot?: Life as it could be
	22	Maryam Masoud Khamis (RUN), Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar
	23	Rianne Conijn (UT), The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging
	24	Lenin da Nobrega Medeiros (VUA/RUN), How are you feeling, human? Towards emotionally supportive chatbots
	25	Xin Du (TUE), The Uncertainty in Exceptional Model Mining
	26	Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization
	27	Ekaterina Muravyeva (TUD), Personal data and informed consent in an educational context
	28	Bibeg Limbu (TUD), Multimodal interaction for deliberate practice: Training complex skills with augmented reality
	29	Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference
	30	Bob Zadok Blok (UL), Creatief, Creatieve, Creatiefst
	31	Gongjin Lan (VU), Learning better – From Baby to Better
	32	Jason Rhuggenaath (TUE), Revenue management in online markets: pricing and online advertising
	33	Rick Gilsing (TUE), Supporting service-dominant business model evaluation in the context of business model innovation
	34	Anna Bon (MU), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development
	35	Siamak Farshidi (UU), Multi-Criteria Decision-Making in Software Production
2021	01	Francisco Xavier Dos Santos Fonseca (TUD),Location-based Games for Social Interaction in Public Space
	02	Rijk Mercuur (TUD), Simulating Human Routines:Integrating Social Practice Theory in Agent-Based Models

03 Seyyed Hadi Hashemi (UVA), Modeling Users Interacting with Smart Devices

04 Ioana Jivet (OU), The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning

05 Davide Dell'Anna (UU), Data-Driven Supervision of Autonomous Systems

06 Daniel Davison (UT), "Hey robot, what do you think?" How children learn with a social robot

07 Armel Lefebvre (UU), Research data management for open science

08 Nardie Fanchamps (OU), The Influence of Sense-Reason-Act Programming on Computational Thinking

09 Cristina Zaga (UT), The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Childrens Collaboration Through Play

10 Quinten Meertens (UvA), Misclassification Bias in Statistical Learning

11 Anne van Rossum (UL), Nonparametric Bayesian Methods in Robotic Vision

12 Lei Pi (UL), External Knowledge Absorption in Chinese SMEs

13 Bob R. Schadenberg (UT), Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning

14 Negin Samaeemofrad (UL), Business Incubators: The Impact of Their Support

15 Onat Ege Adali (TU/e), Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm

16 Esam A. H. Ghaleb (UM), BIMODAL EMOTION RECOGNITION FROM AUDIO-VISUAL CUES

17 Dario Dotti (UM), Human Behavior Understanding from motion and bodily cues using deep neural networks

18 Remi Wieten (UU), Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks

19 Roberto Verdecchia (VU), Architectural Technical Debt: Identification and Management

20 Masoud Mansoury (TU/e), Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems

21 Pedro Thiago Timb Holanda (CWI), Progressive Indexes

22 Sihang Qiu (TUD), Conversational Crowdsourcing

23 Hugo Manuel Proena (LIACS), Robust rules for prediction and description

24 Kaijie Zhu (TU/e), On Efficient Temporal Subgraph Query Processing

25 Eoin Martino Grua (VUA), The Future of E-Health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications

26 Benno Kruit (CWI & VUA), Reading the Grid: Extending Knowledge Bases from Human-readable Tables

27 Jelte van Waterschoot (UT), Personalized and Personal Conversations: Designing Agents Who Want to Connect With You

28 Christoph Selig (UL), Understanding the Heterogeneity of Corporate Entrepreneurship Programs

2022 1 Judith van Stegeren (UT), Flavor text generation for role-playing video games

2 Paulo da Costa (TU/e), Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey

3 Ali el Hassouni (VUA), A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare

4 Ünal Aksu (UU), A Cross-Organizational Process Mining Framework

5 Shiwei Liu (TU/e), Sparse Neural Network Training with In-Time Over-Parameterization

06 Reza Refaei Afshar (TU/e), Machine Learning for Ad Publishers in Real Time Bidding

07 Sambit Praharaj (OU), Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics

08 Maikel L. van Eck (TU/e), Process Mining for Smart Product Design

09 Oana Andreea Inel (VUA), Understanding Events: A Diversity-driven Human-Machine Approach

10 Felipe Moraes Gomes (TUD), Examining the Effectiveness of Collaborative Search Engines

11 Mirjam de Haas (UT), Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers' engagement with robots and tasks during second-language tutoring

12 Guanyi Chen (UU), Computational Generation of Chinese Noun Phrases

13 Xander Wilcke (VUA), Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented Knowledge

14 Michiel Overeem (UU), Evolution of Low-Code Platforms

15 Jelmer Jan Koorn (UU), Work in Process: Unearthing Meaning using Process Mining

16 Pieter Gijsbers (TU/e), Systems for AutoML Research

17 Laura van der Lubbe (VUA), Empowering vulnerable people with serious games and gamification

18 Paris Mavromoustakos Blom (TiU), Player Affect Modelling and Video Game Personalisation

19 Bilge Yigit Ozkan (UU), Cybersecurity Maturity Assessment and Standardisation

20 Fakhra Jabeen (VUA), Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media

21 Seethu Mariyam Christopher (UM), Intelligent Toys for Physical and Cognitive Assessments

22 Alexandra Sierra Rativa (TiU), Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations

23 Ilir Kola (TUD), Enabling Social Situation Awareness in Support Agents

24 Samaneh Heidari (UU), Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values

25 Anna L.D. Latour (LU), Optimal decision-making under constraints and uncertainty

26 Anne Dirkson (LU), Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences

27 Christos Athanasiadis (UM), Emotion-aware cross-modal domain adaptation in video sequences

28 Onuralp Ulusoy (UU), Privacy in Collaborative Systems

29 Jan Kolkmeier (UT), From Head Transform to Mind Transplant: Social Interactions in Mixed Reality

30 Dean De Leo (CWI), Analysis of Dynamic Graphs on Sparse Arrays

31 Konstantinos Traganos (TU/e), Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management

32 Cezara Pastrav (UU), Social simulation for socio-ecological systems

	33	Brinn Hekkelman (CWI/TUD), Fair Mechanisms for Smart Grid Congestion Management
	34	Nimat Ullah (VUA), Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change
	35	Mike E.U. Ligthart (VUA), Shaping the Child-Robot Relationship: Interaction Design Patterns for a Sustainable Interaction
2023	01	Bojan Simoski (VUA), Untangling the Puzzle of Digital Health Interventions
	02	Mariana Rachel Dias da Silva (TiU), Grounded or in flight? What our bodies can tell us about the whereabouts of our thoughts
	03	Shabnam Najafian (TUD), User Modeling for Privacy-preserving Explanations in Group Recommendations
	04	Gineke Wiggers (UL), The Relevance of Impact: bibliometric-enhanced legal information retrieval
	05	Anton Bouter (CWI), Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization, Including Real-World Medical Applications
	06	António Pereira Barata (UL), Reliable and Fair Machine Learning for Risk Assessment
	07	**Tianjin Huang** (TU/e), The Roles of Adversarial Examples on Trustworthiness of Deep Learning